

Documentation | EN

EPP6001-0002

1-channel serial interface (RS232 / RS422 / RS485)



Table of contents

| | | |
|----------|--|-----------|
| 1 | Foreword | 5 |
| 1.1 | Notes on the documentation | 5 |
| 1.2 | Safety instructions | 6 |
| 1.3 | Documentation Issue Status | 7 |
| 2 | Product group: EtherCAT P Box modules | 8 |
| 3 | Product overview | 9 |
| 3.1 | Introduction | 9 |
| 3.2 | Technical data | 10 |
| 3.3 | Scope of supply..... | 11 |
| 3.4 | Process image | 12 |
| 3.4.1 | Control word..... | 14 |
| 3.4.2 | Status word | 14 |
| 3.5 | Technology..... | 15 |
| 4 | Mounting and connection | 16 |
| 4.1 | Mounting | 16 |
| 4.1.1 | Dimensions | 16 |
| 4.1.2 | Fixing..... | 17 |
| 4.1.3 | Functional earth (FE) | 17 |
| 4.1.4 | Tightening torques for plug connectors..... | 17 |
| 4.2 | EtherCAT P | 18 |
| 4.2.1 | Connectors..... | 19 |
| 4.2.2 | Status LEDs | 20 |
| 4.2.3 | Conductor losses | 21 |
| 4.3 | RS232 | 22 |
| 4.3.1 | Connector..... | 22 |
| 4.3.2 | Status LEDs | 22 |
| 4.4 | RS422 | 23 |
| 4.4.1 | Connectors..... | 23 |
| 4.4.2 | Status LEDs | 23 |
| 4.5 | RS485 | 24 |
| 4.5.1 | Connector..... | 24 |
| 4.5.2 | Status LEDs | 24 |
| 4.6 | Digital inputs/outputs | 25 |
| 4.6.1 | Connector..... | 25 |
| 4.6.2 | Status LEDs | 25 |
| 4.6.3 | Examples of external connections | 26 |
| 4.7 | UL Requirements | 27 |
| 4.8 | Disposal | 28 |
| 5 | Commissioning/Configuration | 29 |
| 5.1 | Integrating into a TwinCAT project..... | 29 |
| 5.2 | Serial interface | 30 |
| 5.2.1 | Setting the interface type | 30 |
| 5.2.2 | Setting the interface parameters | 34 |

| | | |
|----------|---|-----------|
| 5.2.3 | Communication by PLC program | 35 |
| 5.2.4 | Communication via a virtual COM port | 37 |
| 5.3 | Digital inputs/outputs | 38 |
| 5.3.1 | Activating process data | 38 |
| 5.3.2 | Assignment of connector pins to process data | 38 |
| 5.4 | CoE objects | 39 |
| 5.4.1 | Directory | 39 |
| 5.4.2 | Object description and parameterization | 40 |
| 5.5 | Restoring the delivery state | 54 |
| 5.6 | Decommissioning | 55 |
| 6 | Appendix | 56 |
| 6.1 | General operating conditions | 56 |
| 6.2 | Accessories | 57 |
| 6.3 | Version identification of EtherCAT devices | 58 |
| 6.3.1 | General notes on marking | 58 |
| 6.3.2 | Version identification of EP/EPI/EPP/ER/ERI boxes | 59 |
| 6.3.3 | Beckhoff Identification Code (BIC) | 60 |
| 6.3.4 | Electronic access to the BIC (eBIC) | 62 |
| 6.4 | Support and Service | 64 |

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement.

No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 with corresponding applications or registrations in various other countries.



EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of instructions

In this documentation the following instructions are used.
These instructions must be read carefully and followed without fail!

DANGER

Serious risk of injury!

Failure to follow this safety instruction directly endangers the life and health of persons.

WARNING

Risk of injury!

Failure to follow this safety instruction endangers the life and health of persons.

CAUTION

Personal injuries!

Failure to follow this safety instruction can lead to injuries to persons.

NOTE

Damage to environment/equipment or data loss

Failure to follow this instruction can lead to environmental damage, equipment damage or data loss.



Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation Issue Status

| Version | Comment |
|---------|---|
| 1.3 | <ul style="list-style-type: none"> • EtherCAT P status LEDs updated |
| 1.2 | <ul style="list-style-type: none"> • Dimensions updated • UL requirements updated |
| 1.1 | <ul style="list-style-type: none"> • Front page updated • Structure update |
| 1.0 | <ul style="list-style-type: none"> • First release |

Firmware and hardware versions

This documentation refers to the firmware and hardware version that was applicable at the time the documentation was written.

The module features are continuously improved and developed further. Modules having earlier production statuses cannot have the same properties as modules with the latest status. However, existing properties are retained and are not changed, so that older modules can always be replaced with new ones.

The firmware and hardware version (delivery state) can be found in the batch number (D-number) printed on the side of the EtherCAT Box.

Syntax of the batch number (D-number)

D: WW YY FF HH

WW - week of production (calendar week)

YY - year of production

FF - firmware version

HH - hardware version

Example with D no. 29 10 02 01:

29 - week of production 29

10 - year of production 2010

02 - firmware version 02

01 - hardware version 01

Further information on this topic: [Version identification of EtherCAT devices \[► 58\]](#).

2 Product group: EtherCAT P Box modules

EtherCAT P

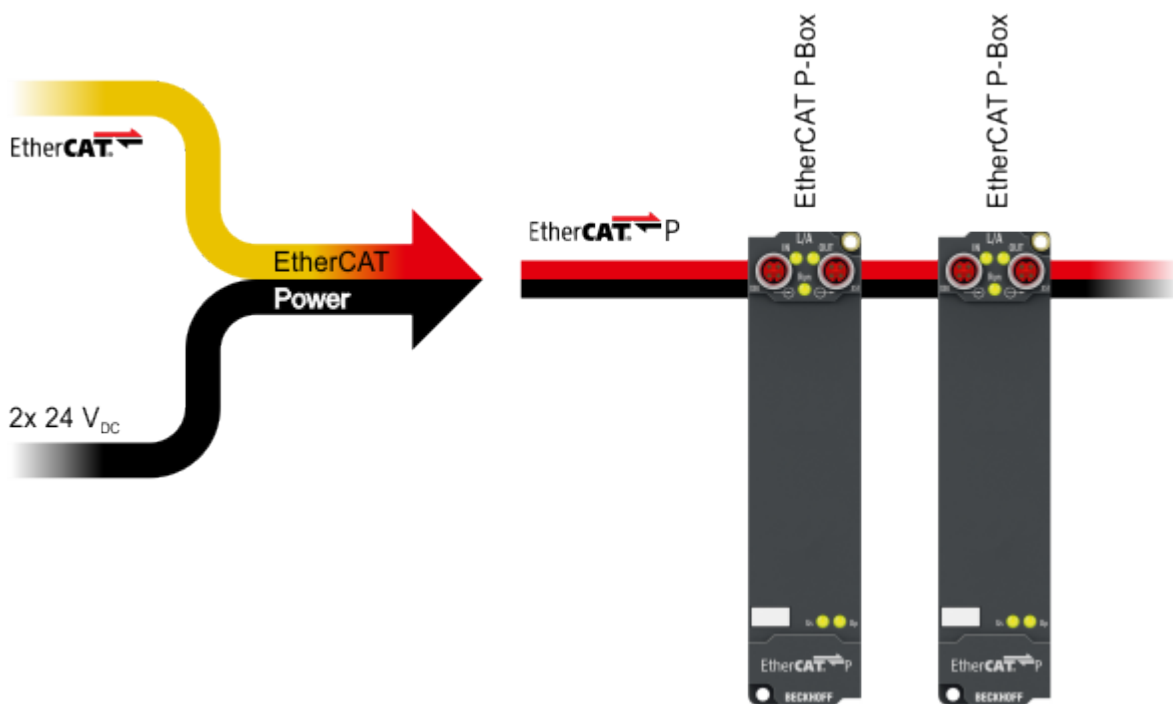
EtherCAT P supplements the EtherCAT technology with a process in which communication and supply voltages are transmitted on a common line. All EtherCAT properties are retained with this process.

Two supply voltages are transmitted per EtherCAT P line. The supply voltages are electrically isolated from each other and can therefore be switched individually. The nominal supply voltage for both is 24 V_{DC}.

EtherCAT P uses the same cable structure as EtherCAT: a 4-core Ethernet cable with M8 connectors. The connectors are mechanically coded so that EtherCAT connectors and EtherCAT P connectors cannot be interchanged.

EtherCAT P Box modules

EtherCAT P Box modules are EtherCAT P slaves with degree of protection IP67. They are designed for operation in wet, dirty or dusty industrial environments.



EtherCAT basics

A detailed description of the EtherCAT system can be found in the [EtherCAT system documentation](#).

3 Product overview

3.1 Introduction

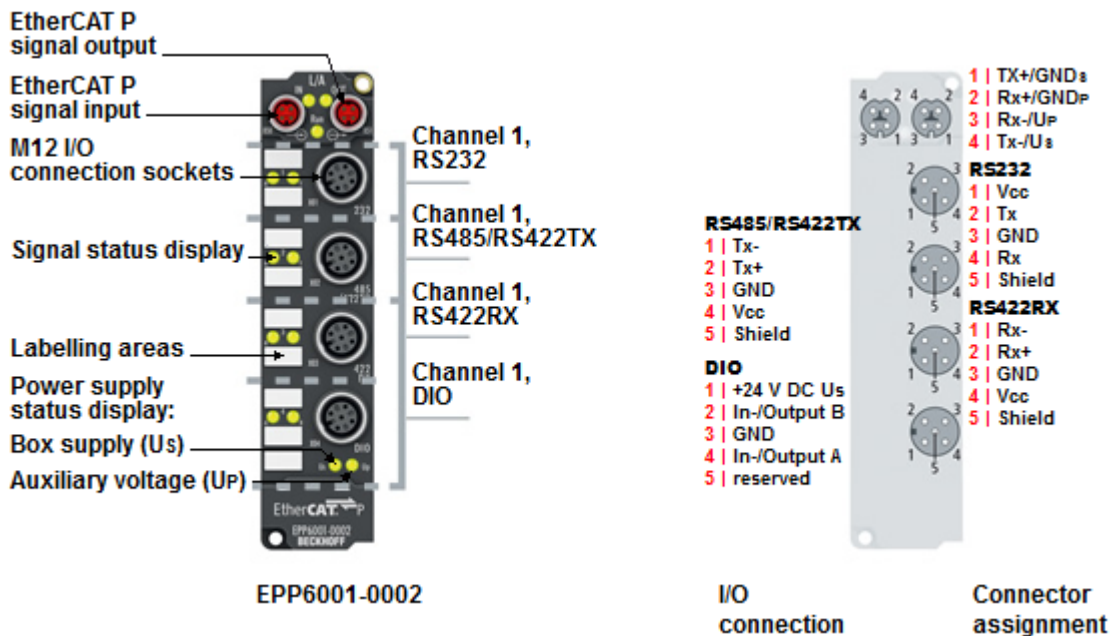


Fig. 1: EPP6001-0002

EPP6001-0002 | 1-channel serial interface, RS232, RS422/RS485

The EPP6001-0002 serial interface module allows the connection of devices with an RS232 or RS422/RS485 interface. The module transmits the data in a fully transparent manner to the higher-level automation device. The active serial communication channel functions independently of the higher-level bus system in full duplex mode at up to 115,200 baud, while a 864 byte receive buffer and a 128 byte send buffer are available. The 1-channel version has an increased end device power supply of up to 1 A, the connector assignment depends on the selected interface. The two integrated digital inputs/outputs allow the connection of additional sensors/actuators in order, for example, to trigger the reading process of the barcode reader or, depending on the result, to initiate an action. In conjunction with the TwinCAT Virtual Serial COM Driver EPP6001-0002 can be used as a normal Windows COM interface.

3.2 Technical data

All values are typical values over the entire temperature range, unless stated otherwise.

| Technical data | EPP6001-0002 |
|---|---|
| Fieldbus | |
| Fieldbus | EtherCAT |
| Connection | EtherCAT P: Combined connection for EtherCAT and supply voltages Input: 1 x M8 socket, 4-pin, P-coded Downstream connection: 1 x M8 socket, 4-pin, P-coded |
| Supply voltages | |
| Connection | See Fieldbus connection |
| Control voltage U_s | |
| Nominal voltage | 24 V _{DC} (-15 % / +20 %) |
| Sum current | max. 3 A ¹⁾ |
| Consumers | <ul style="list-style-type: none"> • Module electronics: 100 mA • Remote communication station (e.g. barcode scanner) • Loads at digital outputs |
| Peripheral voltage U_p | |
| Nominal voltage | 24 V _{DC} (-15 % / +20 %) |
| Sum current | max. 3 A ¹⁾ |
| Current consumption from U_p | None. U_p is only forwarded. |
| Serial interface | |
| Number of channels | 1 |
| Interface type | Parameterizable: <ul style="list-style-type: none"> • RS232 • RS422 • RS485 |
| Connection | RS232: 1x M12 socket RS422: 2x M12 socket RS485: 1x M12 socket |
| End device supply voltage ²⁾ | 5 V _{DC} from the control voltage U_s max. 1 A, short-circuit proof |
| Cable length | RS232: max. 15 m RS422/RS485: max. 1,000 m |
| Data transfer rate | Parameterizable: 300 .. 115,200 baud (bit/s) |
| Data format | Parameterizable: 8N1, 7E1, 7O1, 8N1, 8E1, 8O1, 7E2, 7O2, 8N2, 8E2, 8O2 |
| Bit distortion | < 3 % |

¹⁾ Sum current of consumers and power transmission.

²⁾ Supply voltage available at the connections of the serial interface.

| Technical data | | EPP6001-0002 |
|--------------------------------------|---|---------------------|
| Digital inputs/outputs | | |
| Number | 2 | |
| Connection | 1x M12 socket | |
| I/O supply voltage ³⁾ | 24 V _{DC} from the control voltage U _s max. 0.5 A, short-circuit proof | |
| Input specification | | |
| Characteristics | Type 3 according to EN 61131-2, compatible with type 1 | |
| Input filter | 10 µs | |
| Signal voltage "0" | -3 .. +5 V _{DC} | |
| Signal voltage "1" | +11 .. +30 V _{DC} | |
| Input current | 3 mA at 24 V _{DC} | |
| Output specification | | |
| Load type | ohmic, inductive, lamp load | |
| Nominal voltage | 24 V _{DC} (-15%/+20%) from the control voltage U _s | |
| Output current | max. 0.5 A per channel, short-circuit proof | |
| Environmental conditions | | |
| Ambient temperature during operation | -25...+60 °C -25...+55 °C according to cULus | |
| Ambient temperature during storage | -40...+85 °C | |
| Vibration / shock resistance | conforms to EN 60068-2-6 / EN 60068-2-27 Additional checks | |
| EMC immunity / emission | conforms to EN 61000-6-2 / EN 61000-6-4 | |
| Protection class | IP65, IP66, IP67 (conforms to EN 60529) | |
| Mechanics | | |
| Weight | approx. 165 g | |
| Installation position | variable | |
| Approvals and conformity | | |
| Approvals | CE, cULus | |

³⁾ Supply voltage available at the connection of the digital inputs/outputs.

3.3 Scope of supply

Make sure that the following components are included in the scope of delivery:

- 1x EtherCAT P Box EPP6001-0002
- 2x protective cap for EtherCAT P socket, M8, red (pre-assembled)
- 10x labels, blank (1 strip of 10)








Pre-assembled protective caps do not ensure IP67 protection

























Protective caps are pre-assembled at the factory to protect connectors during transport. They may not be tight enough to ensure IP67 protection.

Ensure that the protective caps are correctly seated to ensure IP67 protection.

3.4 Process image

- ▲  Term 1 (EPP6001-0002)
 - ▷  COM Inputs
 - ▷  COM Outputs
 - ▷  WcState
 - ▷  InfoData

COM Inputs

- ▲  COM Inputs
 - ▶  Status
 - ▶  Data In 0
 - ▶  Data In 1
 - ▶  Data In 2
 - ▶  Data In 3
 - ▶  Data In 4
 - ▶  Data In 5
 - ▶  Data In 6
 - ▶  Data In 7
 - ▶  Data In 8
 - ▶  Data In 9
 - ▶  Data In 10
 - ▶  Data In 11
 - ▶  Data In 12
 - ▶  Data In 13
 - ▶  Data In 14
 - ▶  Data In 15
 - ▶  Data In 16
 - ▶  Data In 17
 - ▶  Data In 18
 - ▶  Data In 19
 - ▶  Data In 20
 - ▶  Data In 21

Status
























Status word for receive data.

Data In [n]

The input variables "Data In 0" .. "Data In 22" each contain one byte of receive data (USINT).

"Data In 0" contains the first-received byte.

COM outputs

- ▲  COM Outputs
 -  Data Out 0
 -  Data Out 1
 -  Data Out 2
 -  Data Out 3
 -  Data Out 4
 -  Data Out 5
 -  Data Out 6
 -  Data Out 7
 -  Data Out 8
 -  Data Out 9
 -  Data Out 10
 -  Data Out 11
 -  Data Out 12
 -  Data Out 13
 -  Data Out 14
 -  Data Out 15
 -  Data Out 16
 -  Data Out 17
 -  Data Out 18
 -  Data Out 19
 -  Data Out 20
 -  Data Out 21

Ctrl

Control word for transmit data.

Data Out [n]

The output variables "Data Out 0" .. "Data Out 22" can each be filled with one byte of send data.

The content of "Data Out 0" is transmitted first.

3.4.1 Control word

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|----|----|----|----|
| Name | OL7 | OL6 | OL5 | OL4 | OL3 | OL2 | OL1 | OL0 | - | - | - | - | SC | IR | RA | TR |

| Bit no. | Name | Description |
|---------|--------------------------|---|
| 15 .. 8 | OL7...OL0 (OutLength) | $1_{dec} \dots 22_{dec}$ The number of output bytes available for the transmission from the controller to the box. |
| 7 .. 4 | reserved | |
| 3 | SC (SendContinuous) | rise Continuous sending of data from the FIFO. The send buffer is filled (up to 128 bytes) by the controller. The buffer content is sent with rising edge of bit SC. Once the data has been transferred, this is acknowledged by the box to the controller by setting the SW.2 bit. SW.2 is cancelled with CW.3. |
| 2 | IR (InitRequest) | 1_{bin} The controller requests the box to initialize. The send and receive functions are blocked, the FIFO pointers are reset, and the interface is initialized with the values of the responsible objects (baud rate 4073, data frame 4074, feature bits 4075). The execution of the initialization is acknowledged by the box with the SW.2 (IA) bit. |
| | | 0_{bin} The controller once again requests the box to prepare for serial data exchange. |
| 1 | RA (ReceiveAccepted) | toggle The controller acknowledges receipt of data by changing the state of this bit. Only then can new data be transferred from the box to the controller. |
| 0 | TR (TransmitRequest) | toggle Via a change of state of this bit the controller notifies the box that the DataOut bytes contain the number of bytes indicated via the OL bits. The box acknowledges receipt of the data in the status byte by changing the state of the SW.0 (TA) bit. Only then can new data be transferred from the controller to the box. |

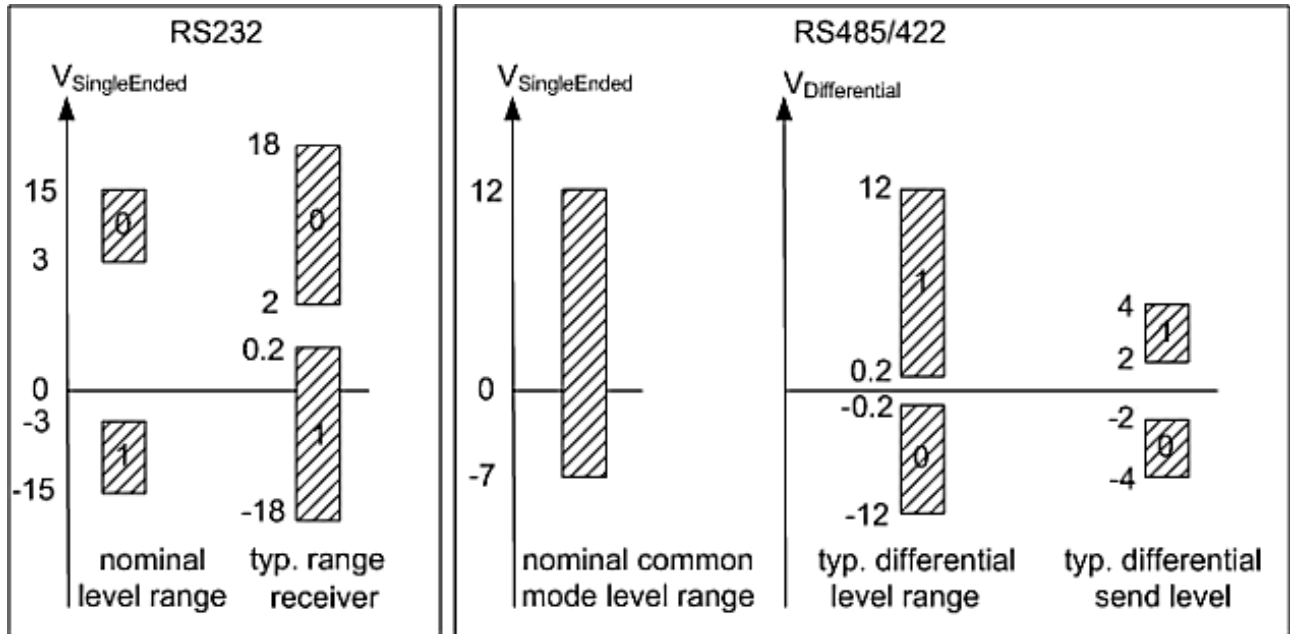
3.4.2 Status word

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|---|----------------|----------------|---------------|-------|----|----|----|
| Name | IL7 | IL6 | IL5 | IL4 | IL3 | IL2 | IL1 | IL0 | - | OVERRUN ERR | FRAMING ERR | PARITY ERR | BUF_F | IA | RR | TA |

Key

| Bit no. | Name | Description |
|----------|---------------------------|--|
| 15 ... 8 | IL7 ... IL0 (InLength) | $1_{dec} \dots 22_{dec}$ The number of input bytes available for the transmission from the box to the controller. |
| 7 | reserved | |
| 6 | OVERRUN ERR | 0 .. 1 An overrun error has occurred. The data concerned is not loaded to the receive FIFO of the box and is lost. |
| 5 | FRAMING ERR | 0 .. 1 A framing error has occurred. The data concerned is not loaded to the receive FIFO of the box and is lost. |
| 4 | PARITY ERR | 0 .. 1 A parity error has occurred. The data concerned is not loaded to the receive FIFO of the box and is lost. |
| 3 | BUF_F | 1 The number of bytes in the receive buffer exceeds the value of parameter 8010:1A "Rx buffer full notification" (factory setting: 864 bytes). |
| 2 | IA (InitAccepted) | 1 The initialization has been executed by the box. |
| | | 0 The box is ready again for serial data exchange. |
| 1 | RR (ReceiveRequest) | toggle Via a change of state of this bit the box notifies the controller that the DataIn bytes contain the number of bytes indicated via the IL bits. The controller has to acknowledge receipt of the data in the control byte via a change of state of bit CW.1 (RA). Only then can new data be transferred from the box to the controller. |
| 0 | TA (TransmitAccepted) | toggle The box acknowledges the receipt of data by changing the state of this bit. Only then can new data be transferred from the controller to the box. |

3.5 Technology



voltages on wire depends on load and cabling

Fig. 2: Level of RS232, RS422, RS485 interfaces

Data transfer rate

The process image contains 22 bytes of user data. It is possible to transmit or receive these 22 bytes every second PLC cycle at the most:

- The data is transferred from the box to the controller in the first PLC cycle.
- In the second PLC cycle, the controller must acknowledge that it has accepted the data.

Therefore, if the cycle time is 10 ms, 50 times 22 bytes can be transmitted per second.

If the data format is set to 8N1, each transmitted byte is made up of a start bit, eight data bits and a stop bit. This is equivalent to 10 bits per byte of user data.

With the above-mentioned settings, a **continuous** data transfer rate of:

- $50[1/s] \times 22[\text{bytes}] \times 10[\text{bits}] = 11000$ baud (bit/s)

can be achieved.

The next lower standard data transfer rate is 9600 baud. Accordingly, continuous transfer at a maximum baud rate of 9600 can be secured with a cycle time of 10 ms.

If only low quantities of data are transmitted or received sporadically (e.g. barcode scanner), the data transfer rate can also be set higher, or the cycle time can be enlarged.

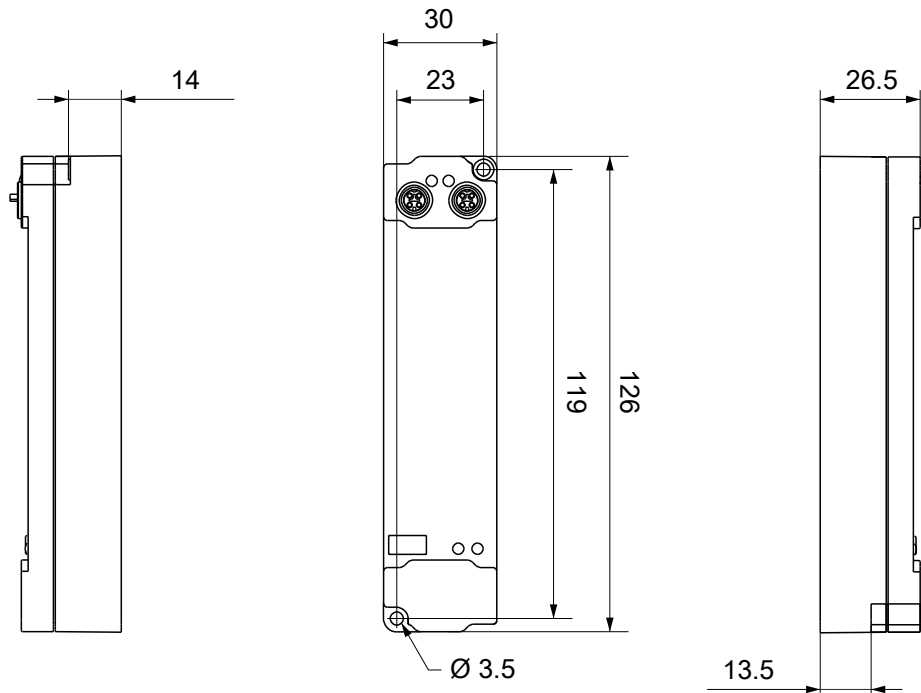
If the controller cannot fetch the data quickly enough from the box, they are buffered in the box's receive buffer. All further data are lost if the receive buffer is full.

A buffer is also available for the send data. With a baud rate of 300 and a data format of 8N1, the box can only transmit 30 bytes per second. However, if more than 30 byte come in per second, the send buffer is written to first in this case also. Once this is full, all further data will be lost.

4 Mounting and connection

4.1 Mounting

4.1.1 Dimensions



All dimensions are given in millimeters.
The drawing is not true to scale.

Housing features

| | |
|------------------------|---|
| Housing material | PA6 (polyamide) |
| Sealing compound | polyurethane |
| Mounting | two mounting holes Ø 3.5 mm for M3 |
| Metal parts | brass, nickel-plated |
| Contacts | CuZn, gold-plated |
| Installation position | variable |
| Protection class | IP65, IP66, IP67 (conforms to EN 60529) when screwed together |
| Dimensions (H x W x D) | approx. 126 x 30 x 26.5 mm (without connectors) |

4.1.2 Fixing

NOTE

Dirt during assembly
 Dirty connectors can lead to malfunctions. Protection class IP67 can only be guaranteed if all cables and connectors are connected.

- Protect the plug connectors against dirt during the assembly.

Mount the module with two M3 screws on the mounting holes in the corners of the module. The mounting holes have no thread.

4.1.3 Functional earth (FE)

The upper mounting holes also serves as a connection for functional earth (FE).

Make sure that the box is grounded to low impedance via the functional earth (FE) connection. You can achieve this, for example, by mounting the box on a grounded machine bed.



Fig. 3: Connection for functional earth (FE)

4.1.4 Tightening torques for plug connectors

Screw connectors tight with a torque wrench. (e.g. ZB8801 from Beckhoff)

| Connector diameter | Tightening torque |
|--------------------|-------------------|
| M8 | 0.4 Nm |
| M12 | 0.6 Nm |

4.2 EtherCAT P

⚠ WARNING

Power supply from SELV/PELV power supply unit!

SELV/PELV circuits (Safety Extra Low Voltage, Protective Extra Low Voltage) according to IEC 61010-2-201 must be used to supply the EtherCAT P Power Sourcing Device (PSD).

Notes:

- SELV/PELV circuits may give rise to further requirements from standards such as IEC 60204-1 et al, for example with regard to cable spacing and insulation.
- A SELV (Safety Extra Low Voltage) supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor, a PELV (Protective Extra Low Voltage) supply also requires a safe connection to the protective conductor.

⚠ CAUTION

Observe the UL requirements

- When operating under UL conditions, observe the warnings in the chapter [UL Requirements](#) [▶ 27].

EtherCAT P transmits two supply voltages:

- **Control voltage U_s**
The following sub-functions are supplied from the control voltage U_s :
 - the fieldbus
 - the processor logic
 - typically the inputs and the sensors if the EtherCAT P Box has inputs.
- **Peripheral voltage U_p**
The digital outputs are typically supplied from the peripheral voltage U_p for EtherCAT P Box modules with digital outputs. U_p can be supplied separately. If U_p is switched off, the fieldbus function, the function of the inputs and the supply of the sensors are maintained.

The exact assignment of U_s and U_p can be found in the pin assignment of the I/O connections.

Redirection of the supply voltages

The supply voltages are passed on internally from the "IN" connection to the "OUT" connection. Hence, the supply voltages U_s and U_p can be passed from one EtherCAT P Box to the next EtherCAT P Box in a simple manner.

NOTE

Note the maximum current.

Ensure that the maximum permitted current of 3 A for the M8 connectors is not exceeded when redirecting EtherCAT P.

4.2.1 Connectors

NOTE

Risk of damage to the device!
 Bring the EtherCAT/EtherCAT P system into a safe, powered down state before starting installation, disassembly or wiring of the modules!

Two M8 sockets at the upper end of the modules are provided for supply and downstream connection of EtherCAT P:

- IN: left M8 socket for EtherCAT P supply
- OUT: right M8 socket for downstream connection of EtherCAT P

The metal threads of the M8 EtherCAT P sockets are internally linked to the FE connection via high impedance RC combination. See chapter [Functional earth \(FE\)](#) [▶ 17].



Fig. 4: Connectors for EtherCAT P

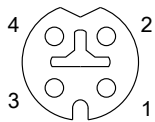


Fig. 5: M8 socket, p-coded

| Contact | Signal | Voltage | Core color ¹⁾ |
|---------|--------|--|--------------------------|
| 1 | Tx + | GND _S | yellow |
| 2 | Rx + | GND _P | white |
| 3 | Rx - | U _P : peripheral voltage, +24 V _{DC} | blue |
| 4 | Tx - | U _S : control voltage, +24 V _{DC} | orange |
| Housing | Shield | Shield | Shield |

¹⁾ The core colors apply to EtherCAT P cables and ECP cables from Beckhoff.

4.2.2 Status LEDs

4.2.2.1 Supply voltage



EtherCAT P Box modules indicate the status of the supply voltages via two status LEDs. The status LEDs are labeled with the designations of the supply voltages: U_s and U_p.

| LED | Display | Meaning |
|--|-------------------|-----------------------------|
| U _s (control voltage) | off | U _s not present. |
| | green illuminated | U _s present. |
| | red illuminated | Error. ¹⁾ |
| U _p (peripheral voltage) | off | U _p not present. |
| | green illuminated | U _p present. |
| | red illuminated | Error. ¹⁾ |

¹⁾ Overload of the sensor supply/auxiliary voltage output at the signal interfaces.
You can find out whether the sensor supply/auxiliary voltage is derived from U_s or from U_p from the assignment of the signal interfaces.

4.2.2.2 EtherCAT



L/A (Link/Act)

A green LED labeled "L/A" or "Link/Act" is located next to each EtherCAT/EtherCAT P socket. The LED indicates the communication state of the respective socket:

| LED | Meaning |
|---------|---|
| off | no connection to the connected EtherCAT device |
| lit | LINK: connection to the connected EtherCAT device |
| flashes | ACT: communication with the connected EtherCAT device |

Run

Each EtherCAT slave has a green LED labelled "Run". The LED signals the status of the slave in the EtherCAT network:

| LED | Meaning |
|----------------------|--------------------------------------|
| off | Slave is in "Init" state |
| flashes uniformly | Slave is in "Pre-Operational" state |
| flashes sporadically | Slave is in "Safe-Operational" state |
| lit | Slave is in "Operational" state |

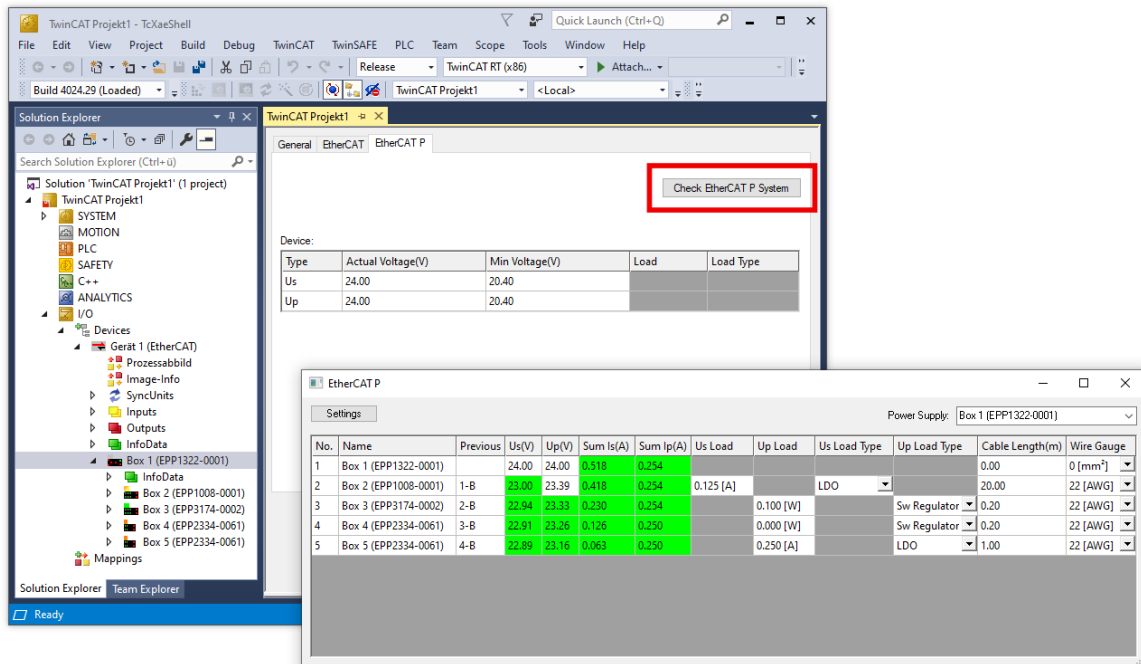
Description of the EtherCAT slave states

4.2.3 Conductor losses

Take into account the voltage drop on the supply line when planning a system. Avoid the voltage drop being so high that the supply voltage at the box lies below the minimum nominal voltage. Variations in the voltage of the power supply unit must also be taken into account.

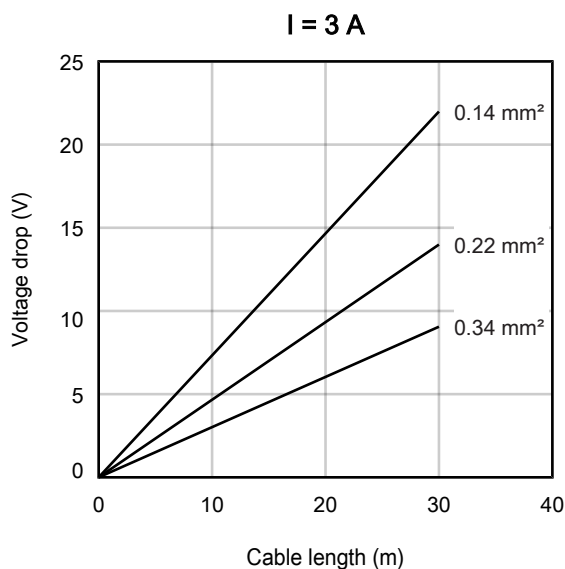
i Planning tool for EtherCAT P

You can plan cable lengths, voltages and currents of your EtherCAT P system using TwinCAT 3. The requirement for this is TwinCAT 3 Build 4020 or higher.



Further information can be found in the quick start guide [IO configuration in TwinCAT](#) in chapter "Configuration of EtherCAT P via TwinCAT".

Voltage drop on the supply line



4.3 RS232

4.3.1 Connector

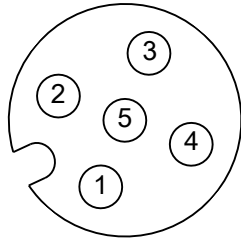


Fig. 6: M12 socket

M12 socket X01

| Pin | Wire color | Signal | Description |
|-----|------------|--------------------|---|
| 1 | brown | VCC | 5 V _{DC} Supply voltage output |
| 2 | white | TxD | Send data |
| 3 | blue | GND _{ISO} | Ground |
| 4 | black | RxD | Receive data |
| 5 | gray | Shield | Shield |

4.3.2 Status LEDs



Fig. 7: RS232 Status LEDs

| LED | Display | Meaning |
|------------|--------------------|--|
| R left | green illuminated | The serial port is ready to receive data. |
| | orange illuminated | The serial port is receiving data. |
| T right | green illuminated | The serial port is ready to transmit data. |
| | orange illuminated | The serial port is transmitting data. |

4.4 RS422

4.4.1 Connectors

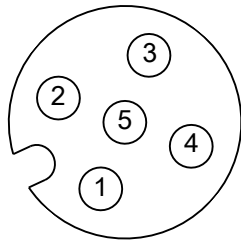


Fig. 8: M12 socket

M12 socket X02

| Pin | Wire color | Signal | Description |
|-----|------------|--------|---------------------------|
| 1 | brown | Tx - | send data |
| 2 | white | Tx + | send data |
| 3 | blue | GND | Ground |
| 4 | black | Vcc | End device supply voltage |
| 5 | grey | Shield | Shield |

M12 socket X03

| Pin | Wire color | Signal | Description |
|-----|------------|--------|---------------------------|
| 1 | brown | Rx - | receive data |
| 2 | white | Rx + | receive data |
| 3 | blue | GND | Ground |
| 4 | black | Vcc | End device supply voltage |
| 5 | grey | Shield | Shield |

4.4.2 Status LEDs



Fig. 9: RS422 Status LEDs

| LED | Display | Meaning |
|------------|--------------------|--|
| R left | green illuminated | The serial port is ready to receive data. |
| | orange illuminated | The serial port is receiving data. |
| T right | green illuminated | The serial port is ready to transmit data. |
| | orange illuminated | The serial port is transmitting data. |

4.5 RS485

4.5.1 Connector

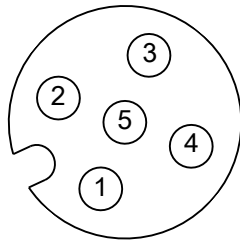


Fig. 10: M12 socket

M12 socket X02

| Pin | Wire color | Signal | Description |
|-----|------------|--------|---------------------------|
| 1 | brown | - / A | Inverted data line |
| 2 | white | + / B | Non-inverted data line |
| 3 | blue | GND | Ground |
| 4 | black | Vcc | End device supply voltage |
| 5 | grey | Shield | Shield |

i The signal designations "A" and "B" are not clear

With some devices the designations of the data lines "A" and "B" are swapped.

i RS485 bus structure - use termination resistors

A linear bus with more than two devices can be set-up in RS485 mode. To prevent reflections during the data transmission, it is necessary to terminate the line ends of the bus cable with resistors (120 Ω).

4.5.2 Status LEDs



Fig. 11: RS485 Status LEDs

| LED | Display | Meaning |
|------------|--------------------|--|
| R left | green illuminated | The serial port is ready to receive data. |
| | orange illuminated | The serial port is receiving data. |
| T right | green illuminated | The serial port is ready to transmit data. |
| | orange illuminated | The serial port is transmitting data. |

4.6 Digital inputs/outputs

4.6.1 Connector

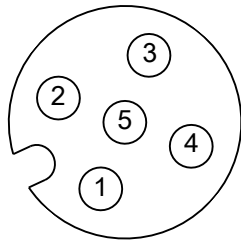


Fig. 12: M12 socket

M12 socket X04

| Pin | Wire color | Signal | Description |
|-----|------------|--------------------|------------------------|
| 1 | brown | 24 V _{DC} | Supply voltage |
| 2 | white | Input/output B | Digital input/output B |
| 3 | blue | GND | Ground |
| 4 | black | Input/output A | Digital input/output A |
| 5 | grey | reserved | - |

4.6.2 Status LEDs



Fig. 13: Status LEDs for digital inputs/outputs

| LED | Display | Meaning |
|------------|-------------------|------------------------------------|
| A left | off | Digital input/output A: Low level |
| | green illuminated | Digital input/output A: High level |
| B right | off | Digital input/output B: Low level |
| | green illuminated | Digital input/output B: High level |

4.6.3 Examples of external connections

M12 socket "4"

The digital input modules acquire the binary control signals from the process level and transmit them to the higher-level automation device.

The signals are connected via M8 connectors (EPxxxx-0001) or M12 connectors (EPxxxx-0002).

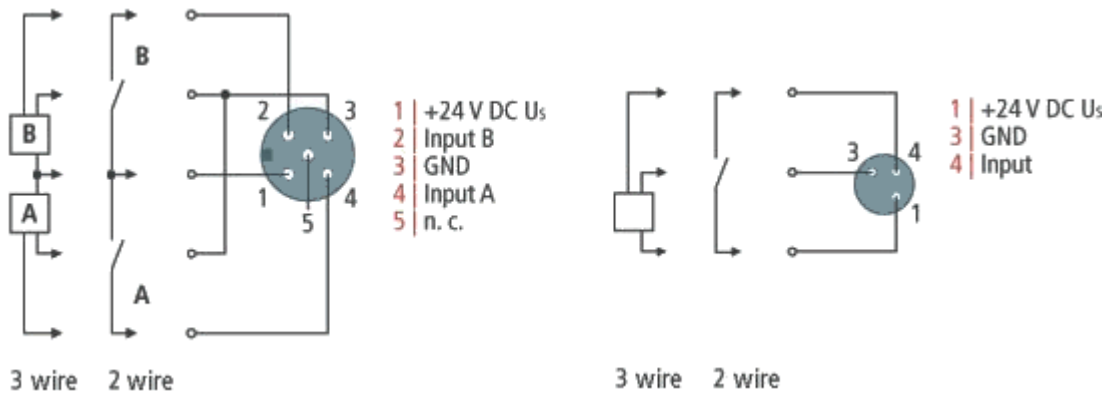


Fig. 14: Digital inputs M8 and M12

The sensors are supplied with a common maximum current of 0.5 A from the control voltage U_s .

Light emitting diodes indicate the signal state of the inputs.

Digital outputs M8 and M12

The digital output modules forward the binary control signals of the automation device to the actuators at the process level.

The signals are connected via M8 connectors (EP2xxx-0001) or M12 connectors (EP2xxx-0002).

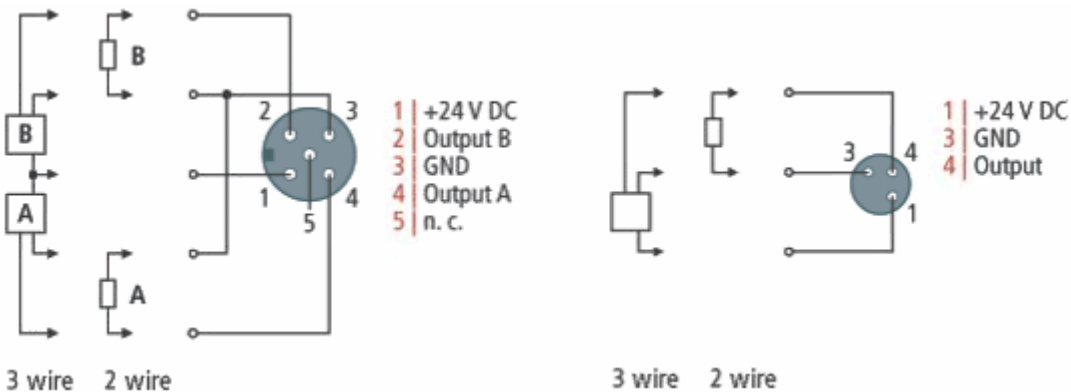


Fig. 15: Digital outputs M8 and M12

The outputs are short-circuit proof and protected against inverse polarity.

LEDs indicate the signal state of the outputs.

4.7 UL Requirements

The installation of the EtherCAT Box Modules certified by UL has to meet the following requirements.

Supply voltage

⚠ CAUTION

CAUTION!

This UL requirements are valid for all supply voltages of all marked EtherCAT Box Modules!
For the compliance of the UL requirements the EtherCAT Box Modules should only be supplied

- by a 24 V_{DC} supply voltage, supplied by an isolating source and protected by means of a fuse (in accordance with UL248), rated maximum 4 Amp, or
- by a 24 V_{DC} power source, that has to satisfy *NEC class 2*.
A *NEC class 2* power supply shall not be connected in series or parallel with another (class 2) power source!

⚠ CAUTION

CAUTION!

To meet the UL requirements, the EtherCAT Box Modules must not be connected to unlimited power sources!

Networks

⚠ CAUTION

CAUTION!

To meet the UL requirements, EtherCAT Box Modules must not be connected to telecommunication networks!

Ambient temperature range

⚠ CAUTION

CAUTION!

To meet the UL requirements, EtherCAT Box Modules has to be operated only at an ambient temperature range of -25 °C to +55 °C!

Marking for UL

All EtherCAT Box Modules certified by UL (Underwriters Laboratories) are marked with the following label.



Fig. 16: UL label

4.8 Disposal



Products marked with a crossed-out wheeled bin shall not be discarded with the normal waste stream. The device is considered as waste electrical and electronic equipment. The national regulations for the disposal of waste electrical and electronic equipment must be observed.

5 Commissioning/Configuration

5.1 Integrating into a TwinCAT project

The procedure for integration in a TwinCAT project is described in these [Quick start guide](#).

5.2 Serial interface

5.2.1 Setting the interface type

Via CoE objects the following settings can be done for the interfaces:

● Parameterization

i The module is parameterized via the "CoE online" tab (with a double-click on the corresponding object). Only the mandatory parameters for the respective interface mode are specified here. Further settings may be possible.

5.2.1.1 RS232

RS232: point-to-point connection to an RS232 device

Direct connection to an RS232 end device, full duplex data transmission (default setting).

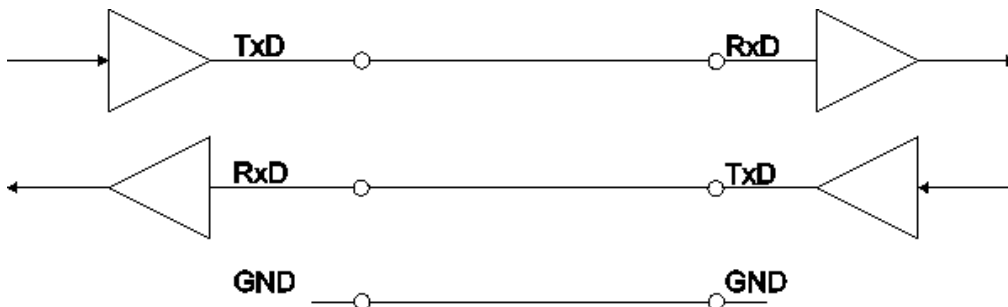


Fig. 17: Point-to-point connection to an RS232 device

The following CoE objects must be set

| Index | Name | Meaning | Data type | Flags | Setting | |
|---------|---------------------|---------|-----------|-------|---------|---------------------------------------|
| F800:01 | Interface Type Ch 1 | 0x00 | RS232 | BIT1 | RW | 0x00 (0 _{dec}) (default) |
| | | 0x01 | RS485/422 | | | |

5.2.1.2 RS422

RS422: 4-wire point-to-point connection to an RS422 device

Direct connection to an RS422 end device, full duplex data transmission.
Data can be transmitted in full duplex in RS422 mode. Only point-to-point connections can be established.

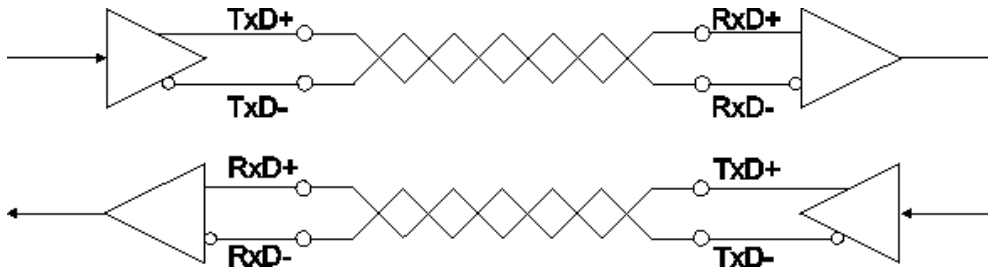


Fig. 18: 4-wire point-to-point connection to an RS422 device

The following CoE objects must be set

| Index | Name | Meaning | Data type | Flags | Setting |
|---------|------------------------|---------|-----------|-------|-----------------------------|
| F800:01 | Interface type Ch 1 | 0x00 | BIT1 | RW | 0x01 (1 _{dec}) |
| | | 0x01 | | | |

| Index | Name | Meaning | Data type | Flags | Setting |
|---------|--|------------------|-----------|-------|------------------|
| 8000:07 | Enable point-to-point connection (RS422) | 0 _{bin} | BOOLEAN | RW | 1 _{bin} |
| | | 1 _{bin} | | | |

5.2.1.3 RS485

You can operate the RS845 communication in two variants:

- Without diagnosis
- With diagnosis of the transmit data [► 33]

RS485: 2-wire connection in bus structure to RS485 device(s)

Bus structure, half duplex data transmission

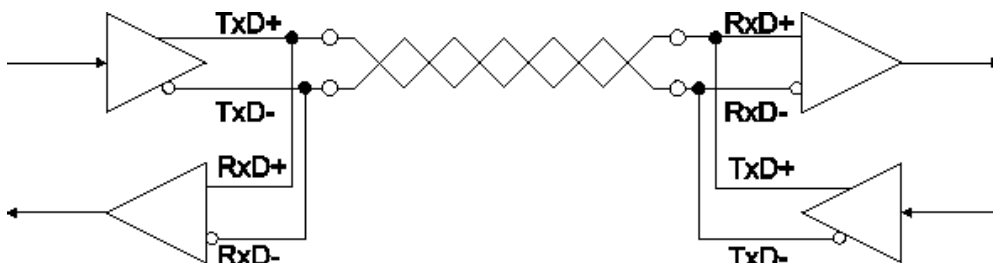


Fig. 19: 2-wire connection in bus structure to RS485 device(s)

The following CoE objects must be set

| Index | Name | Meaning | Data type | Flags | Setting |
|---------|------------------------|---------|-----------|-------|-----------------------------|
| F800:01 | Interface type Ch 1 | 0x00 | BIT1 | RW | 0x01 (1 _{dec}) |
| | | 0x01 | | | |

| Index | Name | Meaning | Data type | Flags | Setting |
|---------|--------------------|------------------|-----------|-------|-------------------------------|
| 8000:06 | Enable half duplex | 0 _{bin} | BOOLEAN | RW | 1 _{bin} (default) |
| | | 1 _{bin} | | | |

| Index | Name | Meaning | Data type | Flags | Setting |
|---------|--|------------------|-----------|-------|------------------|
| 8000:07 | Enable point-to-point connection (RS422) | 0 _{bin} | BOOLEAN | RW | 0 _{bin} |
| | | 1 _{bin} | | | |

i Deactivated receive driver

The receive driver is deactivated during the transmission procedure. The transmitted data are not monitored!

RS485: 2-wire connection with external bridge in bus structure to RS485 device(s)

Bus structure, half duplex data transmission with diagnosis of the transmitted data

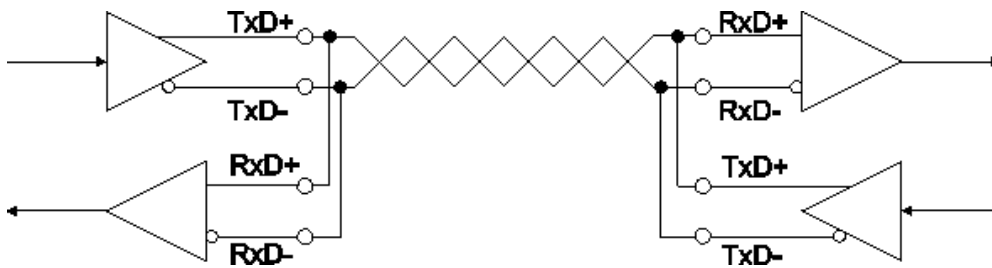


Fig. 20: 2-wire connection with external bridge in bus structure to RS485 device(s)

The following CoE objects must be set

| Index | Name | Meaning | Data type | Flags | Setting |
|---------|------------------------|---------|-----------|-------|-----------------------------|
| F800:01 | Interface type Ch 1 | 0x00 | BIT1 | RW | 0x01 (1 _{dec}) |
| | | 0x01 | | | |

| Index | Name | Meaning | Data type | Flags | Setting |
|---------|--------------------|------------------|-----------|-------|------------------|
| 8000:06 | Enable half duplex | 0 _{bin} | BOOLEAN | RW | 0 _{bin} |
| | | 1 _{bin} | | | |

| Index | Name | Meaning | Data type | Flags | Setting |
|---------|--|------------------|-----------|-------|------------------|
| 8000:07 | Enable point-to-point connection (RS422) | 0 _{bin} | BOOLEAN | RW | 0 _{bin} |
| | | 1 _{bin} | | | |

i Activated receive driver (from firmware version 03)

The receive driver remains activated during the transmission procedure. The transmitted data are monitored! A conditional diagnosis of the line is thus possible. If there is a discrepancy between the transmitted data and the monitored data, it may be assumed that a further receiver also cannot receive these data flawlessly. In this case, check the bus line!

5.2.2 Setting the interface parameters

The parameterization of the module can be set in the CoE (CAN over EtherCAT) list.

i Parameterization via the CoE list (CAN over EtherCAT)

Please note the following general CoE notes when using/manipulating the CoE parameters: - Keep a startup list if components have to be replaced - Differentiation between online/offline dictionary, existence of current XML description - Use "CoE reload" for resetting changes

The following CoE settings are possible from object 0x8000 and are shown below in their default settings:

| Index | Name | Flags | Value |
|---------|--|-------|-------------------|
| 8000:0 | COM Settings Ch.1 | RW | > 27 < |
| 8000:02 | Enable XON/XOFF supported tx data | RW | FALSE |
| 8000:03 | Enable XON/XOFF supported rx data | RW | FALSE |
| 8000:04 | Enable send FIFO data continuous | RW | FALSE |
| 8000:05 | Enable transfer rate optimization | RW | TRUE |
| 8000:06 | Enable half duplex | RW | TRUE |
| 8000:07 | Enable point to point connection (RS422) | RW | FALSE |
| 8000:11 | Baudrate | RW | 9600 Baud (6) |
| 8000:15 | Data frame | RW | 8N1 (3) |
| 8000:19 | Sensor Power Supply Output | RW | 5V (1) |
| 8000:1A | Rx buffer full notification | RW | 0x0360 (864) |
| 8000:1B | Explicit Baudrate | RW | 0x00002580 (9600) |

Fig. 21: CoE settings on object 0x8000 (default)

Continuous transmission of data

A continuous data stream is indispensable for many applications. For this purpose, the Beckhoff modules feature the "Enable send FIFO data continuous" setting in the Settings object. The internal transmit buffer of the box can be filled first by setting this switch. After that the entire contents of the buffer can be transmitted without interruption. To this end, data will be sent from the controller to the box as in a normal transmission. The data from the buffer is only sent with a rising edge of the "Send continuous" bit. If the data has been transferred, the box informs the controller by setting the "Init accepted" bit. "Init accepted" is cleared with "SendContinuous".

Optimization of transfer rates

In normal operating mode the data received will be adopted immediately into the process image. In order to enable a contiguous data stream, the "Enable transfer rate optimization" option in the Settings object is activated by default. Due to this switch, the data will first be stored intermediately in the receive buffer (864 bytes).

The data will only be copied into the process image if no further character is received for 16 bit times or if the buffer is full.

5.2.3 Communication by PLC program

Initialization

Initialization is performed prior to the first transmission/reception. The module is thereby parameterized with the data from the corresponding Settings object.

Procedure:

1. Set "Init request" to 1
 - ⇒ The module confirms successful initialization by setting "Init accepted".
 2. Reset "Init request"
 - ⇒ The module sets "Init accepted" to 0.
- ⇒ The module is ready for data exchange.

Sending data

1. Write the data to be sent in the output variables Data Out [n] [▶ 13].
2. Set the *Output Length* parameter in the Control word to the number of bytes to be transmitted.
3. Toggle the *Transmit Request* bit in the Control word.
 - ⇒ The module acknowledges the data transmission in the Status word via the *Transmit Accepted* parameter.

Receiving data

If the module in the Status word toggles the *Receive Request* bit, there are new receive data in the process data.

1. Read the *Input Length* parameter from the Status word. It contains the number of bytes to be received.
 - ⇒ The data are located in the input variable Data In [n]. The first-received data is located in Data In 0.
2. After reading the data, acknowledge this by toggling the *Receive Accepted* bit in the Control word.
 - Only after that does the module transfer new data from the receive buffer to the process data.

Prioritization

Since received data normally cannot be repeated by the transmitter, they have a higher priority in the module than data to be transmitted.

Furthermore, the priority decreases as the channel number increases. Hence, the reception of data on channel 1 has the highest priority.

5.2.3.1 Samples

Data transmission from the controller to the module (send 2 characters)

1. Set "Output length" to 2
2. Fill "Data Out 0" and "Data Out 1" with user data
3. Change the state of "Transmit request"
 - ⇒ The module acknowledges receipt by changing the state of the "Transmit accepted" bit.

Data transmission from the module to the controller (receive characters)

1. The module indicates that there is new data in the process image by changing the state of the "Receive request" bit.
2. The number of bytes received is written in "Input length"
3. The controller acknowledges acceptance of the bytes by changing the state of "Receive request".

5.2.4 Communication via a virtual COM port

Application Note DK9322-0411-0041 describes the communication via a virtual COM port, taking the EP6002-0002 as an example.

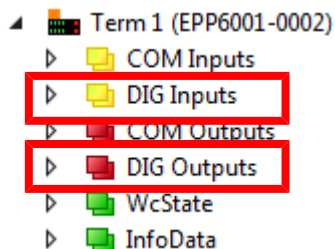
5.3 Digital inputs/outputs

5.3.1 Activating process data

In order to be able to use the digital inputs/outputs, you have to activate the input and output variables in the process image:

✓ Requirement: An EPP6001-0002 has been added in the Solution Explorer under the "I/O" entry.

1. Double-click on the EP6001-0002 IO module.
 2. Click on the **Process Data** tab.
 3. Click on the "Outputs" entry in the **Sync Manager** field.
 4. Activate the **PDO Assignment (0x1C12)** checkbox next to the entry "0x1608".
 - ⇒ The process data object "DIG Outputs", appears in the process data.
 5. Click on the "Inputs" entry in the **Sync Manager** field.
 6. Activate the **PDO Assignment (0x1C13)** checkbox next to the entry "0x1A08".
 - ⇒ The process data object "DIG Inputs", appears in the process data.
- ⇒ Result: The process data objects for the digital inputs/outputs are activated:



5.3.2 Assignment of connector pins to process data

| Connector | Pin | Channel designation | Input variable | Output variable |
|--------------|-----|---------------------|-------------------------------|---------------------------------|
| X04 "DIO" | 2 | B | DIG Inputs Digital Input 2 | DIG Outputs Digital Output 2 |
| | 4 | A | DIG Inputs Digital Input 1 | DIG Outputs Digital Output 1 |

5.4 CoE objects

5.4.1 Directory

| Index (hex) | Name |
|-------------|---|
| 1000 | Device type [► 42] |
| 1008 | Device name [► 42] |
| 1009 | Hardware version [► 42] |
| 100A | Software version [► 42] |
| 1011 | Restore default parameters [► 40] |
| 1018 | Identity [► 42] |
| 10F0 | Backup parameter handling [► 42] |
| 1400 | COM RxPDO-Par Outputs [► 42] |
| 1404 | COM RxPDO-Par Outputs [► 42] |
| 1600 | COM RxPDO-Map Outputs [► 43] |
| 1604 | COM RxPDO-Map Outputs [► 44] |
| 1608 | DIG RxPDO-Map Outputs [► 44] |
| 1800 | COM TxPDO-Par Inputs [► 45] |
| 1804 | COM TxPDO-Par Inputs [► 45] |
| 1A00 | COM TxPDO-Map Inputs [► 46] |
| 1A04 | COM TxPDO-Map Inputs [► 47] |
| 1A08 | DIG TxPDO-Map Inputs [► 47] |
| 1C00 | Sync manager type [► 48] |
| 1C12 | RxPDO assign [► 48] |
| 1C13 | TxPDO assign [► 48] |
| 1C32 | SM output parameter [► 49] |
| 1C33 | SM input parameter [► 50] |
| 6000 | COM Inputs Ch. 1 [► 51] |
| 6001 | Status Ch. 1 [► 51] |
| 6010 | DIG Inputs [► 51] |
| 7000 | COM Outputs Ch. 1 [► 52] |
| 7001 | Ctrl Ch. 1 [► 52] |
| 7010 | DIG Outputs [► 52] |
| 8000 | COM Settings Ch. 1 [► 41] |
| A000 | COM Diag data Ch. 1 [► 53] |
| F000 | Modular device profile [► 53] |
| F008 | Code word [► 53] |
| F010 | Module list [► 53] |
| F800 | COM Settings [► 41] |

5.4.2 Object description and parameterization

● EtherCAT XML Device Description



The display matches that of the CoE objects from the EtherCAT XML Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

● Parameterization via the CoE list (CAN over EtherCAT)



The EtherCAT device is parameterized via the CoE-Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs). Please note the following general CoE notes when using/manipulating the CoE parameters:

- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary, existence of current XML description
- use “CoE reload” for resetting changes

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization [▶ 40] during commissioning
- Objects required for the selection of the interface type [▶ 41]
- Objects intended for regular operation, e.g. through ADS access
- Objects for indicating internal settings [▶ 41] (may be fixed)
- Further profile-specific objects [▶ 50] indicating inputs, outputs and status information

The following section first describes the objects required for normal operation, followed by a complete overview of missing objects.

Objects to be parameterized during commissioning

Index 1011 Restore default parameters

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|----------------------------|---|-----------|-------|--------------------------------|
| 1011:0 | Restore default parameters | Restore default settings | UINT8 | RO | 0x01 (1 _{dec}) |
| 1011:01 | SubIndex 001 | If this object is set to "0x64616F6C" in the set value dialog, all backup objects are reset to their delivery state. | UINT32 | RW | 0x00000000 (0 _{dec}) |

Index 8000 COM Settings Ch.1

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|--|--|-----------|-------|-----------------------------------|
| 8000:0 | COM Settings Ch.1 | | UINT8 | RO | 0x1A (26 _{dec}) |
| 8000:02 | Enable XON/XOFF supported tx data | XON/XOFF is supported for send data | BOOLEAN | RW | 0x00 (0 _{dec}) |
| 8000:03 | Enable XON/XOFF supported rx data | XON/XOFF is supported for receive data | BOOLEAN | RW | 0x00 (0 _{dec}) |
| 8000:04 | Enable send FIFO data continuous | Continuous sending of data from the FIFO enabled | BOOLEAN | RW | 0x00 (0 _{dec}) |
| 8000:05 | Enable transfer rate optimization | Switch on the transfer rate optimization | BOOLEAN | RW | 0x01 (1 _{dec}) |
| 8000:06 | Enable half duplex | Half duplex for RS485 mode (this bit is not evaluated in RS232 and RS422 mode) | BOOLEAN | RW | 0x00 (0 _{dec}) |
| | | 0 Full duplex: The module monitors its transmitted data. | | | |
| | | 1 Half duplex: The module does not monitor the data that it has transmitted itself. | | | |
| 8000:07 | Enable point to point connection (RS422) | 0 The module is used in a bus structure in accordance with the RS485 standard. | BOOLEAN | RW | 0x00 (0 _{dec}) |
| | | 1 The module is used as a point-to-point connection (RS422) | | | |
| 8000:11 | Baud rate | Baud Rate | BIT4 | RW | 0x06 (6 _{dec}) |
| | | 0x01 300 baud | | | |
| | | 0x02 600 baud | | | |
| | | 0x03 1200 baud | | | |
| | | 0x04 2400 baud | | | |
| | | 0x05 4800 baud | | | |
| | | 0x06 9600 baud | | | |
| | | 0x07 19200 baud | | | |
| | | 0x08 38400 baud | | | |
| | | 0x09 57600 baud | | | |
| | | 0x0A 115200 baud | | | |
| 8000:15 | Data frame | Data frame / Stop bits | BIT4 | RW | 0x03 (3 _{dec}) |
| | | 0x01 7E1 | | | |
| | | 0x02 7O1 | | | |
| | | 0x03 8N1 | | | |
| | | 0x04 8E1 | | | |
| | | 0x05 8O1 | | | |
| | | 0x09 7E2 | | | |
| | | 0x0A 7O2 | | | |
| | | 0x0B 8N2 | | | |
| | | 0x0C 8E2 | | | |
| | | 0x0D 8O2 | | | |
| | | 8000:19 | | | |
| 8000:1A | Rx buffer full notification | The value specifies the number of data in the receive FIFO, from which the bit "buffer full" is set. | UINT16 | RW | 0x0360 (864 _{dec}) |
| 8000:1B | Explicit baudrate | In this object the desired baud rate can be entered directly as a number. | UINT32 | RW | 0x00002580 (9600 _{dec}) |

Index F800 COM Settings

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|---------------------|------------|-----------|-------|--------------------------|
| F800:0 | COM Settings | | UINT8 | RO | 0x01 (1 _{dec}) |
| F800:01 | Interface Type Ch 1 | 0x00 RS232 | BIT1 | RW | 0x00 (0 _{dec}) |

Additional objects

Standard objects (0x1000-0x1FFF)

The standard objects have the same meaning for all EtherCAT slaves.

Index 1000 Device type

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-------------|---|-----------|-------|-----------------------------------|
| 1000:0 | Device type | Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile. | UINT32 | RO | 0x00001389 (5001 _{dec}) |

Index 1008 Device name

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-------------|-----------------------------------|-----------|-------|--------------|
| 1008:0 | Device name | Device name of the EtherCAT slave | STRING | RO | EPP6001-0002 |

Index 1009 Hardware version

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|------------------|--|-----------|-------|---------|
| 1009:0 | Hardware version | Hardware version of the EtherCAT slave | STRING | RO | 00 |

Index 100A Software Version

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|------------------|--|-----------|-------|---------|
| 100A:0 | Software version | Firmware version of the EtherCAT slave | STRING | RO | 00 |

Index 1018 Identity

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|---------------|---|-----------|-------|--|
| 1018:0 | Identity | Information for identifying the slave | UINT8 | RO | 0x04 (4 _{dec}) |
| 1018:01 | Vendor ID | Vendor ID of the EtherCAT slave | UINT32 | RO | 0x00000002 (2 _{dec}) |
| 1018:02 | Product code | Product code of the EtherCAT slave | UINT32 | RO | 0x17714052 (393298002 _{dec}) |
| 1018:03 | Revision | Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description | UINT32 | RO | 0x00100002 (1048578 _{dec}) |
| 1018:04 | Serial number | Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0 | UINT32 | RO | 0x00000000 (0 _{dec}) |

Index 10F0 Backup parameter handling

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|---------------------------|---|-----------|-------|--------------------------------|
| 10F0:0 | Backup parameter handling | Information for standardized loading and saving of backup entries | UINT8 | RO | 0x01 (1 _{dec}) |
| 10F0:01 | Checksum | Checksum across all backup entries of the EtherCAT slave | UINT32 | RO | 0x00000000 (0 _{dec}) |

Index 1400 COM RxPDO-Par Outputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------------------|--|-----------------|-------|--------------------------|
| 1400:0 | COM RxPDO-Par Outputs | PDO Parameter RxPDO 1 | UINT8 | RO | 0x06 (6 _{dec}) |
| 1400:06 | Exclude RxPDOs | Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 1 | OCTET-STRING[2] | RO | 04 16 |

Index 1404 COM RxPDO-Par Outputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------------------|--|-----------------|-------|--------------------------|
| 1404:0 | COM RxPDO-Par Outputs | PDO Parameter RxPDO 2 | UINT8 | RO | 0x06 (6 _{dec}) |
| 1404:06 | Exclude RxPDOs | Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 5 | OCTET-STRING[2] | RO | 00 16 |

Index 1600 COM RxPDO-Map Outputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------------------|--|-----------|-------|---------------------------|
| 1600:0 | COM RxPDO-Map Outputs | PDO Mapping RxPDO 1 | UINT8 | RO | 0x1C (28 _{dec}) |
| 1600:01 | SubIndex 001 | 1. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x01 (Transmit request)) | UINT32 | RO | 0x7000:01, 1 |
| 1600:02 | SubIndex 002 | 2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x02 (Receive accepted)) | UINT32 | RO | 0x7000:02, 1 |
| 1600:03 | SubIndex 003 | 3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x03 (Init request)) | UINT32 | RO | 0x7000:03, 1 |
| 1600:04 | SubIndex 004 | 4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x04 (Send continuous)) | UINT32 | RO | 0x7000:04, 1 |
| 1600:05 | SubIndex 005 | 5. PDO Mapping entry (4 bits align) | UINT32 | RO | 0x0000:00, 4 |
| 1600:06 | SubIndex 006 | 6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x09 (Output length)) | UINT32 | RO | 0x7000:09, 8 |
| 1600:07 | SubIndex 007 | 7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0)) | UINT32 | RO | 0x7000:11, 8 |
| 1600:08 | SubIndex 008 | 8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1)) | UINT32 | RO | 0x7000:12, 8 |
| 1600:09 | SubIndex 009 | 9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2)) | UINT32 | RO | 0x7000:13, 8 |
| 1600:0A | SubIndex 010 | 10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3)) | UINT32 | RO | 0x7000:14, 8 |
| 1600:0B | SubIndex 011 | 11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4)) | UINT32 | RO | 0x7000:15, 8 |
| 1600:0C | SubIndex 012 | 12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5)) | UINT32 | RO | 0x7000:16, 8 |
| 1600:0D | SubIndex 013 | 13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6)) | UINT32 | RO | 0x7000:17, 8 |
| 1600:0E | SubIndex 014 | 14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7)) | UINT32 | RO | 0x7000:18, 8 |
| 1600:0F | SubIndex 015 | 15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8)) | UINT32 | RO | 0x7000:19, 8 |
| 1600:10 | SubIndex 016 | 16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9)) | UINT32 | RO | 0x7000:1A, 8 |
| 1600:11 | SubIndex 017 | 17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10)) | UINT32 | RO | 0x7000:1B, 8 |
| 1600:12 | SubIndex 018 | 18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11)) | UINT32 | RO | 0x7000:1C, 8 |
| 1600:13 | SubIndex 019 | 19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12)) | UINT32 | RO | 0x7000:1D, 8 |
| 1600:14 | SubIndex 020 | 20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13)) | UINT32 | RO | 0x7000:1E, 8 |
| 1600:15 | SubIndex 021 | 21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14)) | UINT32 | RO | 0x7000:1F, 8 |
| 1600:16 | SubIndex 022 | 22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15)) | UINT32 | RO | 0x7000:20, 8 |
| 1600:17 | SubIndex 023 | 23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16)) | UINT32 | RO | 0x7000:21, 8 |
| 1600:18 | SubIndex 024 | 24. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17)) | UINT32 | RO | 0x7000:22, 8 |
| 1600:19 | SubIndex 025 | 25. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18)) | UINT32 | RO | 0x7000:23, 8 |
| 1600:1A | SubIndex 026 | 26. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19)) | UINT32 | RO | 0x7000:24, 8 |
| 1600:1B | SubIndex 027 | 27. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20)) | UINT32 | RO | 0x7000:25, 8 |
| 1600:1C | SubIndex 028 | 28. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21)) | UINT32 | RO | 0x7000:26, 8 |

Index 1604 COM RxPDO-Map Outputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------------------|--|-----------|-------|---------------------------|
| 1604:0 | COM RxPDO-Map Outputs | PDO Mapping RxPDO 2 | UINT8 | RO | 0x17 (23 _{dec}) |
| 1604:01 | SubIndex 001 | 1. PDO Mapping entry (object 0x7001 (Ctrl Ch.1), entry 0x01 (Ctrl)) | UINT32 | RO | 0x7001:01, 16 |
| 1604:02 | SubIndex 002 | 2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0)) | UINT32 | RO | 0x7000:11, 8 |
| 1604:03 | SubIndex 003 | 3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1)) | UINT32 | RO | 0x7000:12, 8 |
| 1604:04 | SubIndex 004 | 4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2)) | UINT32 | RO | 0x7000:13, 8 |
| 1604:05 | SubIndex 005 | 5. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3)) | UINT32 | RO | 0x7000:14, 8 |
| 1604:06 | SubIndex 006 | 6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4)) | UINT32 | RO | 0x7000:15, 8 |
| 1604:07 | SubIndex 007 | 7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5)) | UINT32 | RO | 0x7000:16, 8 |
| 1604:08 | SubIndex 008 | 8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6)) | UINT32 | RO | 0x7000:17, 8 |
| 1604:09 | SubIndex 009 | 9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7)) | UINT32 | RO | 0x7000:18, 8 |
| 1604:0A | SubIndex 010 | 10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8)) | UINT32 | RO | 0x7000:19, 8 |
| 1604:0B | SubIndex 011 | 11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9)) | UINT32 | RO | 0x7000:1A, 8 |
| 1604:0C | SubIndex 012 | 12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10)) | UINT32 | RO | 0x7000:1B, 8 |
| 1604:0D | SubIndex 013 | 13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11)) | UINT32 | RO | 0x7000:1C, 8 |
| 1604:0E | SubIndex 014 | 14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12)) | UINT32 | RO | 0x7000:1D, 8 |
| 1604:0F | SubIndex 015 | 15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13)) | UINT32 | RO | 0x7000:1E, 8 |
| 1604:10 | SubIndex 016 | 16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14)) | UINT32 | RO | 0x7000:1F, 8 |
| 1604:11 | SubIndex 017 | 17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15)) | UINT32 | RO | 0x7000:20, 8 |
| 1604:12 | SubIndex 018 | 18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16)) | UINT32 | RO | 0x7000:21, 8 |
| 1604:13 | SubIndex 019 | 19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17)) | UINT32 | RO | 0x7000:22, 8 |
| 1604:14 | SubIndex 020 | 20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18)) | UINT32 | RO | 0x7000:23, 8 |
| 1604:15 | SubIndex 021 | 21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19)) | UINT32 | RO | 0x7000:24, 8 |
| 1604:16 | SubIndex 022 | 22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20)) | UINT32 | RO | 0x7000:25, 8 |
| 1604:17 | SubIndex 023 | 23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21)) | UINT32 | RO | 0x7000:26, 8 |

Index 1608 DIG RxPDO-Map Outputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------------------|---|-----------|-------|--------------------------|
| 1608:0 | DIG RxPDO-Map Outputs | PDO Mapping RxPDO 3 | UINT8 | RO | 0x03 (3 _{dec}) |
| 1608:01 | SubIndex 001 | 1. PDO Mapping entry (object 0x7010 (DIG Outputs), entry 0x01 (Digital Output 1)) | UINT32 | RO | 0x7010:01, 1 |
| 1608:02 | SubIndex 002 | 2. PDO Mapping entry (object 0x7010 (DIG Outputs), entry 0x02 (Digital Output 2)) | UINT32 | RO | 0x7010:02, 1 |
| 1608:03 | SubIndex 003 | 3. PDO Mapping entry (14 bits align) | UINT32 | RO | 0x0000:00, 14 |

Index 1800 COM TxPDO-Par Inputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|----------------------|--|-----------------|-------|--------------------------|
| 1800:0 | COM TxPDO-Par Inputs | PDO parameter TxPDO 1 | UINT8 | RO | 0x06 (6 _{dec}) |
| 1800:06 | Exclude TxPDOs | Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 1 | OCTET-STRING[2] | RO | 04 1A |

Index 1804 COM TxPDO-Par Inputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|----------------------|--|-----------------|-------|--------------------------|
| 1804:0 | COM TxPDO-Par Inputs | PDO parameter TxPDO 2 | UINT8 | RO | 0x06 (6 _{dec}) |
| 1804:06 | Exclude TxPDOs | Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 2 | OCTET-STRING[2] | RO | 00 1A |

Index 1A00 COM TxPDO-Map Inputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|----------------------|--|-----------|-------|---------------------------|
| 1A00:0 | COM TxPDO-Map Inputs | PDO Mapping TxPDO 1 | UINT8 | RO | 0x1F (31 _{dec}) |
| 1A00:01 | SubIndex 001 | 1. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x01 (Transmit accepted)) | UINT32 | RO | 0x6000:01, 1 |
| 1A00:02 | SubIndex 002 | 2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x02 (Receive request)) | UINT32 | RO | 0x6000:02, 1 |
| 1A00:03 | SubIndex 003 | 3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x03 (Init accepted)) | UINT32 | RO | 0x6000:03, 1 |
| 1A00:04 | SubIndex 004 | 4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x04 (Buffer full)) | UINT32 | RO | 0x6000:04, 1 |
| 1A00:05 | SubIndex 005 | 5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x05 (Parity error)) | UINT32 | RO | 0x6000:05, 1 |
| 1A00:06 | SubIndex 006 | 6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x06 (Framing error)) | UINT32 | RO | 0x6000:06, 1 |
| 1A00:07 | SubIndex 007 | 7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x07 (Overrun error)) | UINT32 | RO | 0x6000:07, 1 |
| 1A00:08 | SubIndex 008 | 8. PDO Mapping entry (1 bits align) | UINT32 | RO | 0x0000:00, 1 |
| 1A00:09 | SubIndex 009 | 9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x09 (Input length)) | UINT32 | RO | 0x6000:09, 8 |
| 1A00:0A | SubIndex 010 | 10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0)) | UINT32 | RO | 0x6000:11, 8 |
| 1A00:0B | SubIndex 011 | 11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1)) | UINT32 | RO | 0x6000:12, 8 |
| 1A00:0C | SubIndex 012 | 12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2)) | UINT32 | RO | 0x6000:13, 8 |
| 1A00:0D | SubIndex 013 | 13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3)) | UINT32 | RO | 0x6000:14, 8 |
| 1A00:0E | SubIndex 014 | 14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4)) | UINT32 | RO | 0x6000:15, 8 |
| 1A00:0F | SubIndex 015 | 15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5)) | UINT32 | RO | 0x6000:16, 8 |
| 1A00:10 | SubIndex 016 | 16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6)) | UINT32 | RO | 0x6000:17, 8 |
| 1A00:11 | SubIndex 017 | 17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7)) | UINT32 | RO | 0x6000:18, 8 |
| 1A00:12 | SubIndex 018 | 18. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8)) | UINT32 | RO | 0x6000:19, 8 |
| 1A00:13 | SubIndex 019 | 19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9)) | UINT32 | RO | 0x6000:1A, 8 |
| 1A00:14 | SubIndex 020 | 20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10)) | UINT32 | RO | 0x6000:1B, 8 |
| 1A00:15 | SubIndex 021 | 21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11)) | UINT32 | RO | 0x6000:1C, 8 |
| 1A00:16 | SubIndex 022 | 22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12)) | UINT32 | RO | 0x6000:1D, 8 |
| 1A00:17 | SubIndex 023 | 23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13)) | UINT32 | RO | 0x6000:1E, 8 |
| 1A00:18 | SubIndex 024 | 24. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14)) | UINT32 | RO | 0x6000:1F, 8 |
| 1A00:19 | SubIndex 025 | 25. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15)) | UINT32 | RO | 0x6000:20, 8 |
| 1A00:1A | SubIndex 026 | 26. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16)) | UINT32 | RO | 0x6000:21, 8 |
| 1A00:1B | SubIndex 027 | 27. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17)) | UINT32 | RO | 0x6000:22, 8 |
| 1A00:1C | SubIndex 028 | 28. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18)) | UINT32 | RO | 0x6000:23, 8 |
| 1A00:1D | SubIndex 029 | 29. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19)) | UINT32 | RO | 0x6000:24, 8 |
| 1A00:1E | SubIndex 030 | 30. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20)) | UINT32 | RO | 0x6000:25, 8 |
| 1A00:1F | SubIndex 031 | 31. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21)) | UINT32 | RO | 0x6000:26, 8 |

Index 1A04 COM TxPDO-Map Inputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|----------------------|--|-----------|-------|---------------------------|
| 1A04:0 | COM TxPDO-Map Inputs | PDO Mapping TxPDO 2 | UINT8 | RO | 0x17 (23 _{dec}) |
| 1A04:01 | SubIndex 001 | 1. PDO Mapping entry (object 0x6001 (Status Ch.1), entry 0x01 (Status)) | UINT32 | RO | 0x6001:01, 16 |
| 1A04:02 | SubIndex 002 | 2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0)) | UINT32 | RO | 0x6000:11, 8 |
| 1A04:03 | SubIndex 003 | 3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1)) | UINT32 | RO | 0x6000:12, 8 |
| 1A04:04 | SubIndex 004 | 4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2)) | UINT32 | RO | 0x6000:13, 8 |
| 1A04:05 | SubIndex 005 | 5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3)) | UINT32 | RO | 0x6000:14, 8 |
| 1A04:06 | SubIndex 006 | 6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4)) | UINT32 | RO | 0x6000:15, 8 |
| 1A04:07 | SubIndex 007 | 7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5)) | UINT32 | RO | 0x6000:16, 8 |
| 1A04:08 | SubIndex 008 | 8. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6)) | UINT32 | RO | 0x6000:17, 8 |
| 1A04:09 | SubIndex 009 | 9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7)) | UINT32 | RO | 0x6000:18, 8 |
| 1A04:0A | SubIndex 010 | 10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8)) | UINT32 | RO | 0x6000:19, 8 |
| 1A04:0B | SubIndex 011 | 11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9)) | UINT32 | RO | 0x6000:1A, 8 |
| 1A04:0C | SubIndex 012 | 12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10)) | UINT32 | RO | 0x6000:1B, 8 |
| 1A04:0D | SubIndex 013 | 13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11)) | UINT32 | RO | 0x6000:1C, 8 |
| 1A04:0E | SubIndex 014 | 14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12)) | UINT32 | RO | 0x6000:1D, 8 |
| 1A04:0F | SubIndex 015 | 15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13)) | UINT32 | RO | 0x6000:1E, 8 |
| 1A04:10 | SubIndex 016 | 16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14)) | UINT32 | RO | 0x6000:1F, 8 |
| 1A04:11 | SubIndex 017 | 17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15)) | UINT32 | RO | 0x6000:20, 8 |
| 1A04:12 | SubIndex 018 | 18. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16)) | UINT32 | RO | 0x6000:21, 8 |
| 1A04:13 | SubIndex 019 | 19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17)) | UINT32 | RO | 0x6000:22, 8 |
| 1A04:14 | SubIndex 020 | 20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18)) | UINT32 | RO | 0x6000:23, 8 |
| 1A04:15 | SubIndex 021 | 21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19)) | UINT32 | RO | 0x6000:24, 8 |
| 1A04:16 | SubIndex 022 | 22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20)) | UINT32 | RO | 0x6000:25, 8 |
| 1A04:17 | SubIndex 023 | 23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21)) | UINT32 | RO | 0x6000:26, 8 |

Index 1A08 DIG TxPDO-Map Inputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|---------------------------|---|-----------|-------|--------------------------|
| 1A08:0 | COM TxPDO-Map Inputs Ch.2 | PDO Mapping TxPDO 6 | UINT8 | RO | 0x03 (3 _{dec}) |
| 1A08:01 | SubIndex 001 | 1. PDO Mapping entry (object 0x6010 (DIG Inputs), entry 0x01 (Digital Input 1)) | UINT32 | RO | 0x6010:01, 1 |
| 1A08:02 | SubIndex 002 | 2. PDO Mapping entry (object 0x6010 (DIG Inputs), entry 0x02 (Digital Input 2)) | UINT32 | RO | 0x6010:02, 1 |
| 1A08:03 | SubIndex 003 | 3. PDO Mapping entry (14 bits align) | UINT32 | RO | 0x0000:00, 14 |

Index 1C00 Sync manager type

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-------------------|---|-----------|-------|--------------------------|
| 1C00:0 | Sync manager type | Using the Sync Managers | UINT8 | RO | 0x04 (4 _{dec}) |
| 1C00:01 | SubIndex 001 | Sync-Manager Type Channel 1: Mailbox Write | UINT8 | RO | 0x01 (1 _{dec}) |
| 1C00:02 | SubIndex 002 | Sync-Manager Type Channel 2: Mailbox Read | UINT8 | RO | 0x02 (2 _{dec}) |
| 1C00:03 | SubIndex 003 | Sync-Manager Type Channel 3: Process Data Write (Outputs) | UINT8 | RO | 0x03 (3 _{dec}) |
| 1C00:04 | SubIndex 004 | Sync-Manager Type Channel 4: Process Data Read (Inputs) | UINT8 | RO | 0x04 (4 _{dec}) |

Index 1C12 RxPDO assign

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|--------------|--|-----------|-------|-------------------------------|
| 1C12:0 | RxPDO assign | PDO Assign Outputs | UINT8 | RW | 0x02 (2 _{dec}) |
| 1C12:01 | Subindex 001 | 1. allocated RxPDO (contains the index of the associated RxPDO mapping object) | UINT16 | RW | 0x1604 (5636 _{dec}) |
| 1C12:02 | Subindex 002 | 2. reserved | UINT16 | RW | -- |

Index 1C13 TxPDO assign

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|--------------|--|-----------|-------|-------------------------------|
| 1C13:0 | TxPDO assign | PDO Assign Inputs | UINT8 | RW | 0x02 (2 _{dec}) |
| 1C13:01 | Subindex 001 | 1. allocated TxPDO (contains the index of the associated TxPDO mapping object) | UINT16 | RW | 0x1A04 (6660 _{dec}) |
| 1C13:02 | Subindex 002 | 2. reserved | UINT16 | RW | -- |

Index 1C32 SM output parameter

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-------------------------|---|-----------|-------|-------------------------------------|
| 1C32:0 | SM output parameter | Synchronization parameters for the outputs | UINT8 | RO | 0x20 (32 _{dec}) |
| 1C32:01 | Sync mode | Current synchronization mode: <ul style="list-style-type: none"> • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event | UINT16 | RW | 0x0000 (0 _{dec}) |
| 1C32:02 | Cycle time | Cycle time (in ns): <ul style="list-style-type: none"> • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time | UINT32 | RW | 0x0003D090 (250000 _{dec}) |
| 1C32:03 | Shift time | Time between SYNC0 event and output of the outputs (in ns, DC mode only) | UINT32 | RO | 0x00000384 (900 _{dec}) |
| 1C32:04 | Sync modes supported | Supported synchronization modes: <ul style="list-style-type: none"> • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 Event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output Shift with SYNC1 Event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▶ 49]) | UINT16 | RO | 0x0001 (1 _{dec}) |
| 1C32:05 | Minimum cycle time | Minimum cycle time (in ns) | UINT32 | RO | 0x0003D090 (250000 _{dec}) |
| 1C32:06 | Calc and copy time | Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only) | UINT32 | RO | 0x00000000 (0 _{dec}) |
| 1C32:07 | Minimum delay time | | UINT32 | RO | 0x00000384 (900 _{dec}) |
| 1C32:08 | Command | <ul style="list-style-type: none"> • 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started <p>The entries 1C32:03 [▶ 49], 1C32:05 [▶ 49], 1C32:06 [▶ 49], 1C32:09 [▶ 49], 1C33:03 [▶ 50], 1C33:06 [▶ 49], 1C33:09 [▶ 50] are updated with the maximum measured values. For a subsequent measurement the measured values are reset</p> | UINT16 | RW | 0x0000 (0 _{dec}) |
| 1C32:09 | Maximum Delay time | Time between SYNC1 event and output of the outputs (in ns, DC mode only) | UINT32 | RO | 0x00000384 (900 _{dec}) |
| 1C32:0B | SM event missed counter | Number of missed SM events in OPERATIONAL (DC mode only) | UINT16 | RO | 0x0000 (0 _{dec}) |
| 1C32:0C | Cycle exceeded counter | Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early) | UINT16 | RO | 0x0000 (0 _{dec}) |
| 1C32:0D | Shift too short counter | Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only) | UINT16 | RO | 0x0000 (0 _{dec}) |
| 1C32:20 | Sync error | The synchronization was not correct in the last cycle (outputs were output too late; DC mode only) | BOOLEAN | RO | 0x00 (0 _{dec}) |

Index 1C33 SM input parameter

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-------------------------|--|-----------|-------|-------------------------------------|
| 1C33:0 | SM input parameter | Synchronization parameters for the inputs | UINT8 | RO | 0x20 (32 _{dec}) |
| 1C33:01 | Sync mode | Current synchronization mode: <ul style="list-style-type: none"> • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available) | UINT16 | RW | 0x0000 (0 _{dec}) |
| 1C33:02 | Cycle time | as 1C32:02 [► 49] | UINT32 | RW | 0x0003D090 (250000 _{dec}) |
| 1C33:03 | Shift time | Time between SYNC0 event and reading of the inputs (in ns, only DC mode) | UINT32 | RO | 0x00000384 (900 _{dec}) |
| 1C33:04 | Sync modes supported | Supported synchronization modes: <ul style="list-style-type: none"> • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 Event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [► 49] or 1C33:08 [► 50]) | UINT16 | RO | 0x0001 (1 _{dec}) |
| 1C33:05 | Minimum cycle time | as 1C32:05 [► 49] | UINT32 | RO | 0x0003D090 (250000 _{dec}) |
| 1C33:06 | Calc and copy time | Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode) | UINT32 | RO | 0x00000000 (0 _{dec}) |
| 1C33:07 | Minimum delay time | | UINT32 | RO | 0x00000384 (900 _{dec}) |
| 1C33:08 | Command | as 1C32:08 [► 49] | UINT16 | RW | 0x0000 (0 _{dec}) |
| 1C33:09 | Maximum Delay time | Time between SYNC1 event and reading of the inputs (in ns, only DC mode) | UINT32 | RO | 0x00000384 (900 _{dec}) |
| 1C33:0B | SM event missed counter | as 1C32:11 [► 49] | UINT16 | RO | 0x0000 (0 _{dec}) |
| 1C33:0C | Cycle exceeded counter | as 1C32:12 [► 49] | UINT16 | RO | 0x0000 (0 _{dec}) |
| 1C33:0D | Shift too short counter | as 1C32:13 [► 49] | UINT16 | RO | 0x0000 (0 _{dec}) |
| 1C33:20 | Sync error | as 1C32:32 [► 49] | BOOLEAN | RO | 0x00 (0 _{dec}) |

Profile-specific objects (0x6000-0xFFFF)

The profile-specific objects have the same meaning for all EtherCAT slaves that support the profile 5001.

Index 6000 COM Inputs Ch.1

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-------------------|--|-----------|-------|---------------------------|
| 6000:0 | COM Inputs Ch.1 | | UINT8 | RO | 0x26 (38 _{dec}) |
| 6000:01 | Transmit accepted | The module acknowledges receipt of data by changing the state of this bit | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 6000:02 | Receive request | By changing the state of this bit, the module informs the controller that the DataIn bytes contain the number of bytes displayed in "Input length" | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 6000:03 | Init Accepted | The initialization is carried out from the terminal | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 6000:04 | Buffer full | The receive FIFO is full | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 6000:05 | Parity error | A parity error has occurred | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 6000:06 | Framing error | A framing error has occurred | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 6000:07 | Overrun error | An overrun error has occurred | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 6000:09 | Input length | Number of input bytes available for transfer from the terminal to the controller | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:11 | Data In 0 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:12 | Data In 1 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:13 | Data In 2 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:14 | Data In 3 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:15 | Data In 4 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:16 | Data In 5 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:17 | Data In 6 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:18 | Data In 7 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:19 | Data In 8 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:1A | Data In 9 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:1B | Data In 10 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:1C | Data In 11 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:1D | Data In 12 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:1E | Data In 13 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:1F | Data In 14 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:20 | Data In 15 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:21 | Data In 16 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:22 | Data In 17 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:23 | Data In 18 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:24 | Data In 19 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:25 | Data In 20 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |
| 6000:26 | Data In 21 | Input data | UINT8 | RO | 0x00 (0 _{dec}) |

Index 6001 Status Ch.1

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-------------|--|-----------|-------|----------------------------|
| 6001:0 | Status Ch.1 | | UINT8 | RO | 0x01 (1 _{dec}) |
| 6001:01 | Status | Status word for compatible process image | UINT16 | RO | 0x0000 (0 _{dec}) |

Index 6010 DIG Inputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------------|---|-----------|-------|--------------------------|
| 6010:0 | DIG Inputs | | UINT8 | RO | 0x02 (2 _{dec}) |
| 6010:01 | Digital Input 1 | | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 6010:02 | Digital Input 1 | | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 6010:03 | Init Accepted | The initialization is carried out from the terminal | BOOLEAN | RO | 0x00 (0 _{dec}) |

Index 7000 COM Outputs Ch.1

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|------------------|---|-----------|-------|---------------------------|
| 7000:0 | COM Outputs Ch.1 | | UINT8 | RO | 0x26 (38 _{dec}) |
| 7000:01 | Transmit request | By changing the state of this bit, the controller informs the terminal that the DataOut bytes contain the number of bytes displayed in "Output length". | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 7000:02 | Receive accepted | The controller acknowledges receipt of data by changing the state of this bit. | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 7000:03 | Init request | The controller requests the module to initialize. | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 7000:04 | Send continuous | Continuous sending of data from the FIFO. | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 7000:09 | Output length | Number of output bytes available for transfer from the controller to the terminal. | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:11 | Data Out 0 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:12 | Data Out 1 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:13 | Data Out 2 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:14 | Data Out 3 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:15 | Data Out 4 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:16 | Data Out 5 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:17 | Data Out 6 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:18 | Data Out 7 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:19 | Data Out 8 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:1A | Data Out 9 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:1B | Data Out 10 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:1C | Data Out 11 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:1D | Data Out 12 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:1E | Data Out 13 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:1F | Data Out 14 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:20 | Data Out 15 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:21 | Data Out 16 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:22 | Data Out 17 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:23 | Data Out 18 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:24 | Data Out 19 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:25 | Data Out 20 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |
| 7000:26 | Data Out 21 | Output data | UINT8 | RO | 0x00 (0 _{dec}) |

Index 7001 Ctrl Ch.1

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------|---|-----------|-------|----------------------------|
| 7001:0 | Ctrl Ch.1 | | UINT8 | RO | 0x01 (1 _{dec}) |
| 7001:01 | Ctrl | Control word for compatible process image | UINT16 | RO | 0x0000 (0 _{dec}) |

Index 7010 DIG Outputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|------------------|---------|-----------|-------|---------------------------|
| 7010:0 | DIG Outputs | | UINT8 | RO | 0x26 (38 _{dec}) |
| 7010:01 | Digital Output 1 | | BOOLEAN | RO | 0x00 (0 _{dec}) |
| 7010:02 | Digital Output 2 | | BOOLEAN | RO | 0x00 (0 _{dec}) |

Index A000 COM Diag data Ch.1

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|------------------------------|--|-----------|-------|----------------------------|
| A000:0 | COM Diag data Ch.1 | | UINT8 | RO | 0x21 (33 _{dec}) |
| A000:01 | Buffer overflow | A buffer overflow has occurred. | BOOLEAN | RO | 0x00 (0 _{dec}) |
| A000:02 | Parity error | A parity error has occurred. | BOOLEAN | RO | 0x00 (0 _{dec}) |
| A000:03 | Framing error | A framing error has occurred. | BOOLEAN | RO | 0x00 (0 _{dec}) |
| A000:04 | Overrun error | An overrun error has occurred. | BOOLEAN | RO | 0x00 (0 _{dec}) |
| A000:05 | Buffer full | The receive FIFO is full. | BOOLEAN | RO | 0x00 (0 _{dec}) |
| A000:11 | Data bytes in send buffer | Number of data bytes in the send FIFO | UINT16 | RO | 0x0000 (0 _{dec}) |
| A000:21 | Data bytes in receive buffer | Number of data bytes in the receive FIFO | UINT16 | RO | 0x0000 (0 _{dec}) |

Index F000 Modular device profile

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|---------------------------|--|-----------|-------|-----------------------------|
| F000:0 | Modular device profile | General information for the modular device profile | UINT8 | RO | 0x02 (2 _{dec}) |
| F000:01 | Module index distance | Index distance of the objects of the individual channels | UINT16 | RO | 0x0010 (16 _{dec}) |
| F000:02 | Maximum number of modules | Number of channels | UINT16 | RO | 0x0002 (2 _{dec}) |

Index F008 Code word

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------|---------|-----------|-------|--------------------------------|
| F008:0 | Code word | | UINT32 | RW | 0x00000000 (0 _{dec}) |

Index F010 Module list

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|--------------|---------|-----------|-------|----------------------------------|
| F010:0 | Module list | | UINT8 | RW | 0x02 (2 _{dec}) |
| F010:01 | SubIndex 001 | | UINT32 | RW | 0x00000258 (600 _{dec}) |
| F010:02 | SubIndex 002 | | UINT32 | RW | 0x00000118 (280 _{dec}) |

5.5 Restoring the delivery state

To restore the delivery state for backup objects in ELxxxx terminals / EPxxxx- and EPPxxxx box modules, the CoE object *Restore default parameters, SubIndex 001* can be selected in the TwinCAT System Manager (Config mode).

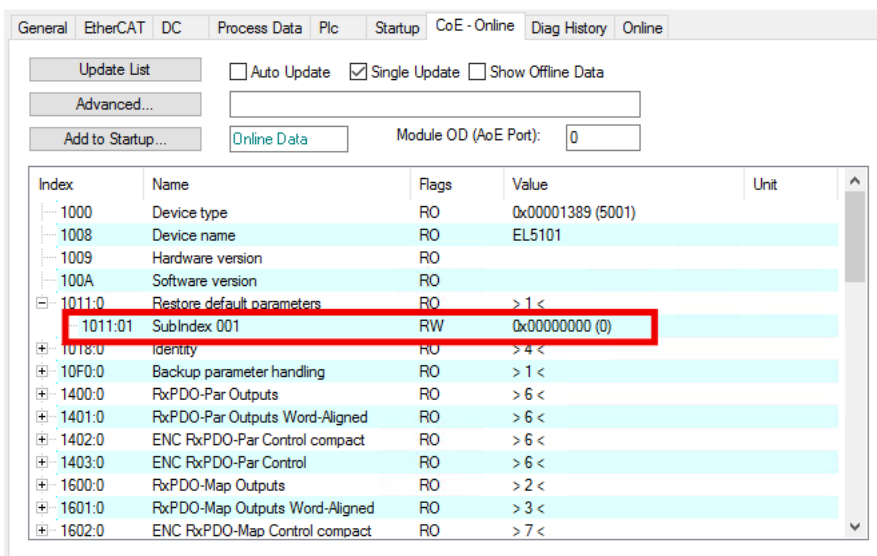


Fig. 22: Selecting the Restore default parameters PDO

Double-click on *SubIndex 001* to enter the Set Value dialog. Enter the value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* and confirm with OK.

All backup objects are reset to the delivery state.

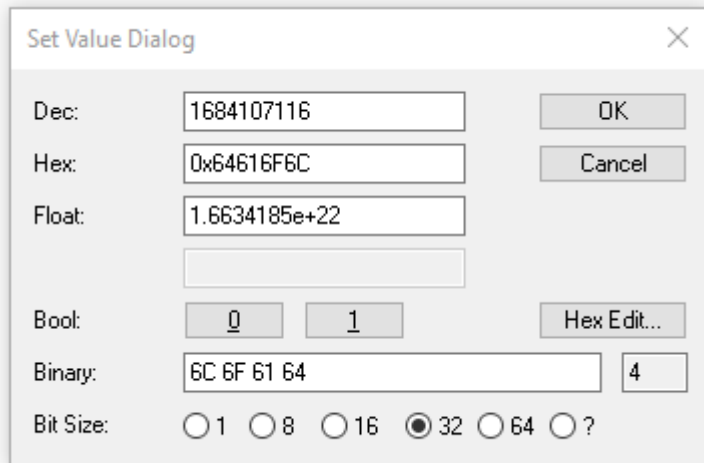


Fig. 23: Entering a restore value in the Set Value dialog

i Alternative restore value

In some older terminals / boxes the backup objects can be switched with an alternative restore value:

Decimal value: 1819238756

Hexadecimal value: 0x6C6F6164

An incorrect entry for the restore value has no effect.

5.6 Decommissioning

⚠ WARNING**Risk of electric shock!**

Bring the bus system into a safe, de-energized state before starting disassembly of the devices!

6 Appendix

6.1 General operating conditions

Protection degrees (IP-Code)

The standard IEC 60529 (DIN EN 60529) defines the degrees of protection in different classes.

| 1. Number: dust protection and touch guard | Definition |
|--|---|
| 0 | Non-protected |
| 1 | Protected against access to hazardous parts with the back of a hand. Protected against solid foreign objects of Ø 50 mm |
| 2 | Protected against access to hazardous parts with a finger. Protected against solid foreign objects of Ø 12.5 mm. |
| 3 | Protected against access to hazardous parts with a tool. Protected against solid foreign objects Ø 2.5 mm. |
| 4 | Protected against access to hazardous parts with a wire. Protected against solid foreign objects Ø 1 mm. |
| 5 | Protected against access to hazardous parts with a wire. Dust-protected. Intrusion of dust is not totally prevented, but dust shall not penetrate in a quantity to interfere with satisfactory operation of the device or to impair safety. |
| 6 | Protected against access to hazardous parts with a wire. Dust-tight. No intrusion of dust. |
| 2. Number: water* protection | Definition |
| 0 | Non-protected |
| 1 | Protected against water drops |
| 2 | Protected against water drops when enclosure tilted up to 15°. |
| 3 | Protected against spraying water. Water sprayed at an angle up to 60° on either side of the vertical shall have no harmful effects. |
| 4 | Protected against splashing water. Water splashed against the disclosure from any direction shall have no harmful effects |
| 5 | Protected against water jets |
| 6 | Protected against powerful water jets |
| 7 | Protected against the effects of temporary immersion in water. Intrusion of water in quantities causing harmful effects shall not be possible when the enclosure is temporarily immersed in water for 30 min. in 1 m depth. |

*) These protection classes define only protection against water.

Chemical Resistance

The Resistance relates to the Housing of the IP67 modules and the used metal parts. In the table below you will find some typical resistance.

| Character | Resistance |
|------------------------------------|---|
| Steam | at temperatures >100°C: not resistant |
| Sodium base liquor (ph-Value > 12) | at room temperature: resistant > 40°C: not resistant |
| Acetic acid | not resistant |
| Argon (technical clean) | resistant |

Key

- resistant: Lifetime several months
- non inherently resistant: Lifetime several weeks
- not resistant: Lifetime several hours resp. early decomposition

6.2 Accessories

Mounting

| Ordering information | Description | Link |
|----------------------|---------------|-------------------------|
| ZS5300-0011 | Mounting rail | Website |

Cables

A complete overview of pre-assembled cables for fieldbus components can be found [here](#).

| Ordering information | Description | Link |
|----------------------|----------------------------------|-------------------------|
| ZK2000-6xxx-xxxx | Sensor cable M12, 4-pin | Website |
| ZK2000-7xxx-0xxx | Sensor cable M12, 4-pin + shield | Website |
| ZK700x-xxxx-xxxx | EtherCAT P cable M8 | Website |

Labeling material, protective caps

| Ordering information | Description |
|----------------------|--|
| ZS5000-0012 | Protective cap for M8 sockets, P-coded, IP67 (50 pieces) |
| ZS5000-0020 | Protective cap for M12 sockets, IP67 (50 pcs.) |
| ZS5100-0000 | Inscription labels, unprinted, 4 strips of 10 |
| ZS5000-xxxx | Printed inscription labels on enquiry |

Tools

| Ordering information | Description |
|----------------------|--|
| ZB8801-0000 | Torque wrench for plugs, 0.4...1.0 Nm |
| ZB8801-0001 | Torque cable key for M8 / wrench size 9 for ZB8801-0000 |
| ZB8801-0002 | Torque cable key for M12 / wrench size 13 for ZB8801-0000 |
| ZB8801-0003 | Torque cable key for M12 field assembly / wrench size 18 for ZB8801-0000 |



Further accessories

Further accessories can be found in the price list for fieldbus components from Beckhoff and online at <https://www.beckhoff.com>.

6.3 Version identification of EtherCAT devices

6.3.1 General notes on marking

Designation

A Beckhoff EtherCAT device has a 14-digit designation, made up of

- family key
- type
- version
- revision

| Example | Family | Type | Version | Revision |
|------------------|--|--|-------------------------------|----------|
| EL3314-0000-0016 | EL terminal (12 mm, non-pluggable connection level) | 3314 (4-channel thermocouple terminal) | 0000 (basic type) | 0016 |
| ES3602-0010-0017 | ES terminal (12 mm, pluggable connection level) | 3602 (2-channel voltage measurement) | 0010 (high-precision version) | 0017 |
| CU2008-0000-0000 | CU device | 2008 (8-port fast ethernet switch) | 0000 (basic type) | 0000 |

Notes

- The elements mentioned above result in the **technical designation**. EL3314-0000-0016 is used in the example below.
- EL3314-0000 is the order identifier, in the case of “-0000” usually abbreviated to EL3314. “-0016” is the EtherCAT revision.
- The **order identifier** is made up of
 - family key (EL, EP, CU, ES, KL, CX, etc.)
 - type (3314)
 - version (-0000)
- The **revision** -0016 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site.
From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. “EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)”.
- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

6.3.2 Version identification of EP/EPI/EPP/ER/ERI boxes

The serial number/ data code for Beckhoff IO devices is usually the 8-digit number printed on the device or on a sticker. The serial number indicates the configuration in delivery state and therefore refers to a whole production batch, without distinguishing the individual modules of a batch.

Structure of the serial number: **KK YY FF HH**

- KK - week of production (CW, calendar week)
- YY - year of production
- FF - firmware version
- HH - hardware version

Example with serial number 12 06 3A 02:

- 12 - production week 12
- 06 - production year 2006
- 3A - firmware version 3A
- 02 - hardware version 02

Exceptions can occur in the **IP67 area**, where the following syntax can be used (see respective device documentation):

Syntax: D ww yy x y z u

- D - prefix designation
- ww - calendar week
- yy - year
- x - firmware version of the bus PCB
- y - hardware version of the bus PCB
- z - firmware version of the I/O PCB
- u - hardware version of the I/O PCB

Example: D.22081501 calendar week 22 of the year 2008 firmware version of bus PCB: 1 hardware version of bus PCB: 5 firmware version of I/O PCB: 0 (no firmware necessary for this PCB) hardware version of I/O PCB: 1

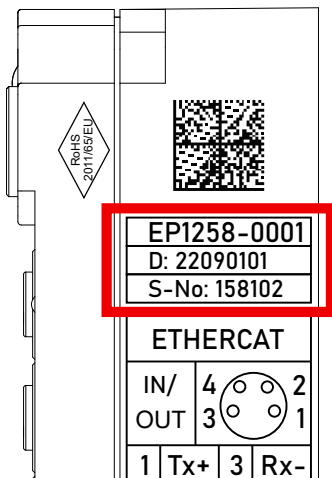


Fig. 24: EP1258-00001 IP67 EtherCAT Box with batch number/DateCode 22090101 and unique serial number 158102

6.3.3 Beckhoff Identification Code (BIC)

The Beckhoff Identification Code (BIC) is increasingly being applied to Beckhoff products to uniquely identify the product. The BIC is represented as a Data Matrix Code (DMC, code scheme ECC200), the content is based on the ANSI standard MH10.8.2-2016.

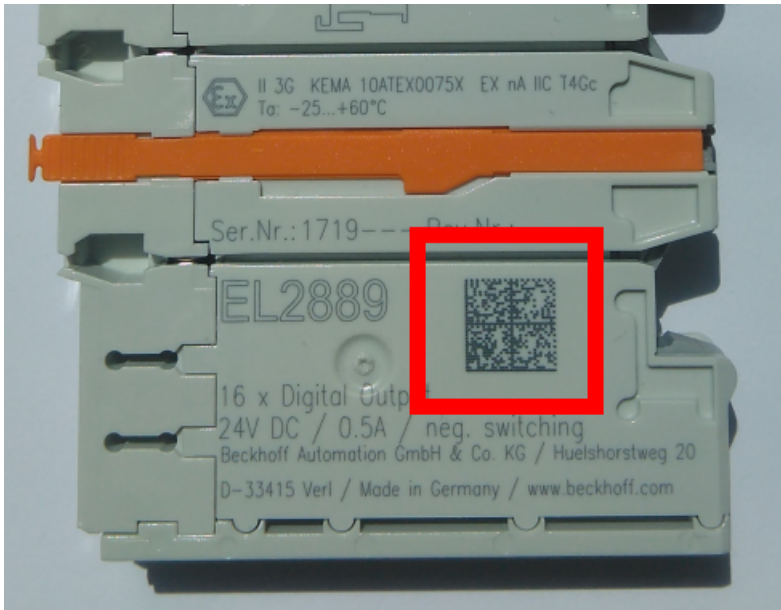


Fig. 25: BIC as data matrix code (DMC, code scheme ECC200)

The BIC will be introduced step by step across all product groups.

Depending on the product, it can be found in the following places:

- on the packaging unit
- directly on the product (if space suffices)
- on the packaging unit and the product

The BIC is machine-readable and contains information that can also be used by the customer for handling and product management.

Each piece of information can be uniquely identified using the so-called data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the table below. If the information is shorter, spaces are added to it.

Following information is possible, positions 1 to 4 are always present, the other according to need of production:

| Position | Type of information | Explanation | Data identifier | Number of digits incl. data identifier | Example |
|----------|------------------------------------|---|-----------------|--|-------------------------|
| 1 | Beckhoff order number | Beckhoff order number | 1P | 8 | 1P 072222 |
| 2 | Beckhoff Traceability Number (BTN) | Unique serial number, see note below | SBTN | 12 | SBTN k4p562d7 |
| 3 | Article description | Beckhoff article description, e.g. EL1008 | 1K | 32 | 1K EL1809 |
| 4 | Quantity | Quantity in packaging unit, e.g. 1, 10, etc. | Q | 6 | Q 1 |
| 5 | Batch number | Optional: Year and week of production | 2P | 14 | 2P 401503180016 |
| 6 | ID/serial number | Optional: Present-day serial number system, e.g. with safety products | 51S | 12 | 51S 678294 |
| 7 | Variant number | Optional: Product variant number on the basis of standard products | 30P | 32 | 30P F971, 2*K183 |
| ... | | | | | |

Further types of information and data identifiers are used by Beckhoff and serve internal processes.

Structure of the BIC

Example of composite information from positions 1 to 4 and with the above given example value on position 6. The data identifiers are highlighted in bold font:

1P072222**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:



Fig. 26: Example DMC **1P**072222**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, position 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTE

This information has been carefully prepared. However, the procedure described is constantly being further developed. We reserve the right to revise and change procedures and documentation at any time and without prior notice. No claims for changes can be made from the information, illustrations and descriptions in this information.

6.3.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

The Beckhoff Identification Code (BIC) is applied to the outside of Beckhoff products in a visible place. If possible, it should also be electronically readable.

Decisive for the electronic readout is the interface via which the product can be electronically addressed.

K-bus devices (IP20, IP67)

Currently, no electronic storage and readout is planned for these devices.

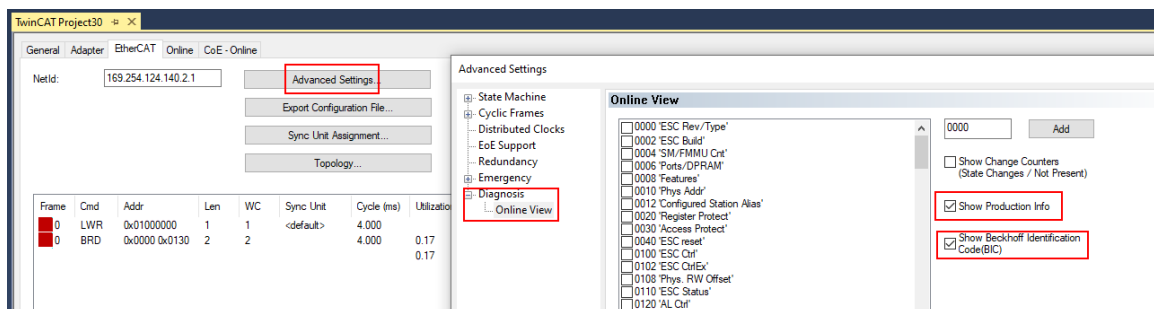
EtherCAT devices (IP20, IP67)

All Beckhoff EtherCAT devices have a so-called ESI-EEPROM, which contains the EtherCAT identity with the revision number. Stored in it is the EtherCAT slave information, also colloquially known as ESI/XML configuration file for the EtherCAT master. See the corresponding chapter in the EtherCAT system manual ([Link](#)) for the relationships.

The eBIC is also stored in the ESI-EEPROM. The eBIC was introduced into the Beckhoff I/O production (terminals, box modules) from 2020; widespread implementation is expected in 2021.

The user can electronically access the eBIC (if existent) as follows:

- With all EtherCAT devices, the EtherCAT master (TwinCAT) can read the eBIC from the ESI-EEPROM
 - From TwinCAT 3.1 build 4024.11, the eBIC can be displayed in the online view.
 - To do this, check the checkbox "Show Beckhoff Identification Code (BIC)" under EtherCAT → Advanced Settings → Diagnostics:



- The BTN and its contents are then displayed:

| No | Addr | Name | State | CRC | Fw | Hw | Production Data | ItemNo | BTN | Description | Quantity | BatchNo | SerialNo |
|----|------|-----------------|-------|-----|----|----|-----------------|--------|----------|-------------|----------|---------|----------|
| 1 | 1001 | Term 1 (EK1100) | OP | 0,0 | 0 | 0 | --- | | | | | | |
| 2 | 1002 | Term 2 (EL1018) | OP | 0,0 | 0 | 0 | 2020 KW36 Fr | 072222 | k4p562d7 | EL1809 | 1 | | 678294 |
| 3 | 1003 | Term 3 (EL3204) | OP | 0,0 | 7 | 6 | 2012 KW24 Sa | | | | | | |
| 4 | 1004 | Term 4 (EL2004) | OP | 0,0 | 0 | 0 | --- | 072223 | k4p562d7 | EL2004 | 1 | | 678295 |
| 5 | 1005 | Term 5 (EL1008) | OP | 0,0 | 0 | 0 | --- | | | | | | |
| 6 | 1006 | Term 6 (EL2008) | OP | 0,0 | 0 | 12 | 2014 KW14 Mo | | | | | | |
| 7 | 1007 | Term 7 (EK1110) | OP | 0 | 1 | 8 | 2012 KW25 Mo | | | | | | |

- Note: as can be seen in the illustration, the production data HW version, FW version and production date, which have been programmed since 2012, can also be displayed with "Show Production Info".
- From TwinCAT 3.1. build 4024.24 the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the Tc2_EtherCAT Library from v3.3.19.0.
- In the case of EtherCAT devices with CoE directory, the object 0x10E2:01 can additionally be used to display the device's own eBIC; the PLC can also simply access the information here:

- The device must be in PREOP/SAFEOP/OP for access:

| Index | Name | Flags | Value |
|---------|---|-------|---|
| 1000 | Device type | RO | 0x015E1389 (22942601) |
| 1008 | Device name | RO | ELM3704-0000 |
| 1009 | Hardware version | RO | 00 |
| 100A | Software version | RO | 01 |
| 100B | Bootloader version | RO | J0.1.27.0 |
| 1011:0 | Restore default parameters | RO | > 1 < |
| 1018:0 | Identity | RO | > 4 < |
| 10E2:0 | Manufacturer-specific Identification C... | RO | > 1 < |
| 10E2:01 | SubIndex 001 | RO | 1P158442SBTN0008jekp1KELM3704 Q1 2P482001000016 |
| 10F0:0 | Backup parameter handling | RO | > 1 < |
| 10F3:0 | Diagnosis History | RO | > 21 < |
| 10F8 | Actual Time Stamp | RO | 0x170bfb277e |

- the object 0x10E2 will be introduced into stock products in the course of a necessary firmware revision.
- From TwinCAT 3.1. build 4024.24 the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the *Tc2_EtherCAT Library* from v3.3.19.0.
- Note: in the case of electronic further processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background
The new BIC information is additionally written as a category in the ESI-EEPROM during the device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored with the help of a category according to ETG.2010. ID 03 indicates to all EtherCAT masters that they must not overwrite these data in case of an update or restore the data after an ESI update.
The structure follows the content of the BIC, see there. This results in a memory requirement of approx. 50..200 bytes in the EEPROM.
- Special cases
 - If multiple, hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC Information.
 - If multiple, non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC Information.
 - If the device consists of several sub-devices with their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

Profibus/Profinet/DeviceNet... Devices

Currently, no electronic storage and readout is planned for these devices.

6.4 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: <https://www.beckhoff.com>

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline: +49 5246 963 157
Fax: +49 5246 963 9157
e-mail: support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline: +49 5246 963 460
Fax: +49 5246 963 479
e-mail: service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone: +49 5246 963 0
Fax: +49 5246 963 198
e-mail: info@beckhoff.com
web: <https://www.beckhoff.com>

More Information:
www.beckhoff.com/epp6001-0002

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

